UNIVERSITY
of
GLASGOW

Optimal Spectral Diagnosis of Hot Solar Plasmas

Scott William McIntosh

Thesis submitted to

the University of Glasgow

for the degree of Ph.D.

in August 1998

Department of Physics & Astronomy,

Kelvin Building,

University of Glasgow,

Glasgow, G12 8QQ.

Abstract

To obtain meaningful diagnostic measurements of hot solar plasmas requires that we must extract the greatest amount of physical information from remotely sensed data whilst differentiating between its information and noise content. The inference of 'reliable' plasma structure models from the data relies heavily upon the inferential 'inversion' method used. Such inversion methods allows us to infer the likely form of the underlying physical source model from the data and theoretical estimates of the emission processes taking place. It is widely known that such 'inverse problems' can give rise to highly ambiguous (non-unique) solutions when errors are present in the observed data. Clearly, an understanding of such inverse approaches and the propagation of errors in data, **and** in the emission rates involved, through to the final solution is paramount in obtaining useful diagnostic measurements. The work presented here addresses inversion formalisms and their application in the face of typical data and emission model uncertainties.

This thesis presents results in a field of study where the uncertainties associated with remote sensing and inverse methodology can run amok if not carefully treated: the inference of the electron density and temperature distribution of the highly inhomogeneous plasmas of the upper solar atmosphere.

In Chapter 1 a brief description is given of the solar atmosphere and why it is best to observe its hotter regions from space. We continue, in Chapter 2, by presenting the necessary theoretical and numerical tools required to understand inverse problems and to make reliable estimates of the underlying plasma structure using such inverse techniques.

Chapter 3 digresses from the main theme to introduce an important data analysis tool which is used extensively in the later chapters of this thesis; the Genetic Algorithm (GA). The flexibility of the GA method is clearly demonstrated therein. As an example we discuss the Gaussian fitting GA (Ga-GA) and its application to the decomposition of real and synthetic emission line spectra.

In Chapter 4 we discuss the ill-posed inference of plasma diagnostic distributions from emission line intensities and ratios. These distributions are widely known as the Differential Emission Measure functions, or DEMs for short. In Section 4.1 we demonstrate that there is a formal relationship between the 'spectroscopic mean values' of n_e, T_e obtained using line ratios and their respective DEM functions $\xi(T_e)$ and $\zeta(n_e)$ with an extension to $\mu(n_e, T_e)$ (the general bivarate DEM function) where mean values of n_e and T_e are simultaneously defined. Following this, in Section 4.2, we develop an entirely novel GA based technique (the Ratio Inversion Technique; RIT), by which we are able to ascertain these diagnostic distributions to a higher degree of uniqueness than methods used previously. In particular, the RIT proves to be quite insensitive to the theoretical uncertainties in the atomic emission models used; which posed a major difficulty in the intensity inversions of previous authors.

In Chapter 5 we present another GA based method (SELECTOR) to overcome the serious numerical instability of inferred DEM functions when noise is present in the observed emission line intensities. We show that the impact of this data noise on the poorly conditioned DEM inversions is dramatically reduced by isolating a subset of emission lines (in the wavelength range of the CDS and SUMER instruments of the ESA/NASA SOlar and Heliospheric Observatory - SOHO - satellite) that improve the conditioning of the DEM inverse problems.

Chapter 6 draws together the points raised and conclusions reached in the preceding chapters and briefly discusses possible improvements, extensions and future applications of the methods introduced.

This study is considered to be both valuable and timely given the increased usage of inverse diagnostics from the high quality data acquired by instruments onboard the aforementioned SOHO satellite.

Contents

List of Figures

List of Tables

Acknowledgements

The task of composing this thesis can only be compared to a relatively short scramble up the north face of the Eiger. There have been *many* sherpas and water-carriers along the way to help me "bag" the summit. I'd like to take this time to thank them all. So, at the risk of sounding like an actor (too late Darren) receiving an Oscar©, here we go :

⋆ My parents and immediate family without whose nurture and support, both financial and emotional, this research would certainly not have been carried out so smoothly. I owe you all too much !

⋆ To my supervisor Prof. John C. Brown for coming up with so many good ideas, supplementing my knowledge with his seemingly boundless insight and for placing me in an atmosphere conducive to work. My "back-up" (take that whatever way you will) supervisor Dr. Declan A. Diver for helping with some of the more tricky aspects of plasma physics and for being there to listen to (and not laugh too loud at) some of my more bizarre ideas. My very special thanks must go to my "surrogate supervisor" Dr. Philip G. Judge of the High Altitude Observatory (HAO; Boulder CO) without whose ability, enthusiasm, moral (and financial) support the vast majority of the results contained in this document could not have evolved. Not forgetting the contribution made by Dr. Paul Charbonneau (also HAO) who went to great lengths to explain to me some of the nastier numerical, statistical and evolutionary aspects of Genetic Algorithms.

⋆ A friendly pat on the back for the two guys who have shared offices with me and subsequently had to put up with my incessant question asking and bad jokes. So thank you David Keston for teaching me about internet protocols (and e^+e^- plasmas) and Martin "Yoda" Hendry for putting up with *all* of my anti-cosmology jibes and trying – "Try not. Do. Or do not. There is no try" – to get to grips with inverse problems at the same time. Also thanks to Susan Morrison and Paul McCallum for being my virtual office, coffee and soul mates.

⋆ All the present and recent "astros", in particular Keith Macpherson, Andrew Conway, Jack Ireland, Richard Barrett and Darren McDonald who have always been around for a chat no matter the triviality of the topic. And, of course, Shashi Kanbur for maintaining the computer system.

⋆ The "lads" for providing some entertainment over the years: George, Craig, Graeme (x2), Kenny and Dave.

⋆ More special thanks must go to Amy Poling and Shannon Jones for helping me through "a little phase" and listening to my tales of woe at a *very* important stage in the completion of this thesis.

This work was undertaken with the support of the UK Particle Physics and Astronomy Research Council (PPARC) using the computing facilities provided by the Starlink project and the generous support of the HAO visitor programme for my valuable research visits in December 1996/1997 and the extended visit during the summer of 1997.

Scott McIntosh, Glasgow

August, 1998

For Alan

I have already explained to you that what is out of the common is usually a guide rather than a hindrance. In solving a problem of this sort, the grand thing is to be able to reason backward. That is a very useful accomplishment, and a very easy one, but people do not practice it much. In the everyday affairs of life it is more useful to reason forward, and so the other comes to be neglected. There are fifty who can reason synthetically for one who can reason analytically.

Sherlock Holmes from "A Study in Scarlet" by A. Conan-Doyle.

Chapter 1

Introduction

This Chapter

This chapter concentrates on the foundations and development of solar UV/EUV emission line spectroscopy. Indeed, through basic models of the solar atmosphere and discussion of the properties of the Earth's atmosphere we discuss the need for remote sensing the hot solar and astrophysical plasmas from space. Some of the first pieces of work detailing the remote sensing of such astrophysical plasmas, derivation of the electron density of planetary nebulae by Menzel et al. (1941) and the 'detection' of the seemingly erroneous temperature of the solar corona by Edlén (1943), show that the basic spectroscopic techniques employed today have remained relatively unchanged. However, the great flux of data from the ESA/NASA Solar and Heliospheric Observatory (SOHO) mission, now entering its third year of operation, a thorough quantitative study of the reliable inference of plasma characteristics and of related theoretical plasma modelling is timely.

In this chapter we introduce some of the particulars of the solar atmosphere and express our motivation for spending vast amounts of time, money and effort in attempting to understand the mysteries it presents us with. The atoms and ions that constitute the Sun's atmosphere emit (and absorb) electromagnetic (e-m) radiation; it is our understanding the basic physical mechanisms that generate this radiation that provide us with clues to understanding of the underlying processes we observe. There is a problem here though, the Earth's atmosphere doesn't make this 'remote sensing'[1] easy. The wavelengths of radiation from hotter regions of the solar atmosphere that are particularly important to understand are almost completely absorbed before reaching ground-based observing sites. This means that we must

[1] In brief, remote sensing is the indirect measurement of the properties of distant and awkward objects.

observe the Sun from the inhospitable reaches of space invariably using unmanned drones. Here we introduce the basic facts about the atmosphere of the Sun, and discuss the need for it to be observed from outwith our protective atmosphere. We briefly discuss some of the landmarks of space-borne solar observing from the use of World War II rocket technology through to the major mission of today, the joint ESA/NASA mission called the SOlar and Heliospheric Observatory or just SOHO[2]. Section 1.3 gives a short overview of the scope and the motivation behind the material that will appear in the following chapters.

1.1 The outer solar atmosphere

The Sun is the sentinel of our region of the Galaxy and it provides us with the energy we need to survive and maintain the state of equilibrium we call life. We are being constantly bombarded by radiation (and particles) emitted by the Sun and the key to understanding the processes happening on the Sun is in 'catching' some of this radiation. To understand the physical mechanisms behind the structure of the Sun's atmosphere is to understand the Sun itself. The radiation emitted (or absorbed) by it tells us of its temperature/density structure, chemical composition, velocity and many other important physical quantities. We will discuss the probing of the Sun's atmosphere[3] in due course but we must first give its physical description.

The goal of this thesis is not in the discussion of particular solar features, i.e. those prominent in all images of the Sun, but of the diagnostic methods employed by the solar physics community to study them. We will consider the atmosphere of the Sun as, to make the discussion simple, a plane-parallel model generated by data from Vernazza et al. (1981) and use it to introduce some solar terminology, see figure 1.1 (noting that we are using a universal reference point as the zero in solar altitude, the point where the vertical optical depth, τ, of the atmosphere at a wavelength of 5000 Å is unity). At low heights (i.e. below 600 kilometers), $T_e(z)$ shows a monotonic decrease as z increases. This region is called the *photosphere* and is where the bulk of the optical (and therefore the bulk of all) radiation is emitted. Higher still in the atmosphere (between 600 and 2,000 kilometers) the temperature gradient changes sign to produce a region with near constant temperature ($4,500 \leq T_e \leq 10,000$ K) this region, called the *chromosphere*, which sees a decrease with height in the electron (and gas) density

[2]Another commonly used acronym is SoHO, but we will use SOHO hereafter.

[3]We use here a working definition for atmosphere, it is "those regions where radiation can escape freely into the surrounding medium".

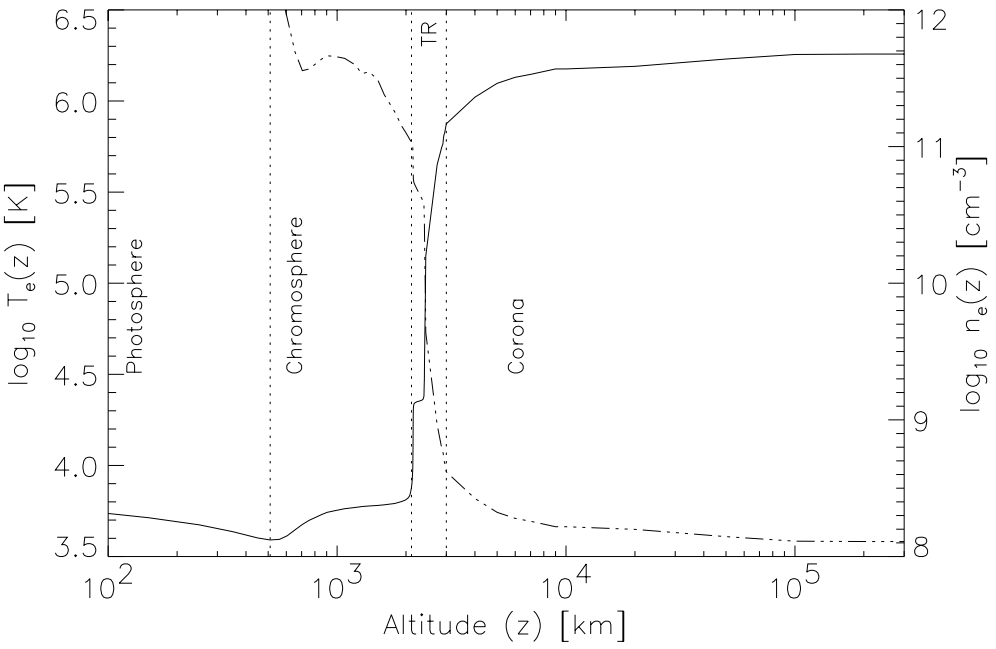

Figure 1.1: Average temperature (solid line) and density (dashed line) structure of the quiet solar photosphere, chromosphere, transition region (TR) and corona. From the photosphere ($z = 0$) to a temperature of 4.5×10^5 K the values plotted are given in Vernazza et al. (1981) and into the corona from a quiet "network" model given in Mariska (1992).

by several orders of magnitude. Yet higher in the atmosphere (above 2,500 kilometers in our model) we enter a region called the *corona* which has an average temperature around $1,000,000$ K. The region between chromosphere and corona is a complex beast, imaginatively named the *transition region* and is the subject of an excellent monograph by Mariska (1992). Indeed, there are other monographs dedicated to specific solar regimes, the photosphere (much of which is discussed in Stenflo 1994), chromosphere (Thomas & Athay 1961; Bray et al. 1984), corona (Golub & Pasachoff 1997) and some magnificent books covering most solar topics and phenomena (Sturrock 1985; Zirin 1988; Stix 1989; Foukal 1990).

By now, even in this very simple description, alarm bells should be ringing because the second law of thermodynamics states that heat cannot (by thermal processes) flow from cooler material to hotter material. So, what mechanism heats the chromosphere and corona and what produces the "discontinuous" jump in temperature of the transition region ? These are a couple of the **big** puzzles of solar physics.

To continue our argument we must digress a little. As noted above, the photosphere

produces the bulk of the optical "white" light but the corona was only (until the advent of space observations) visible[4] briefly during solar eclipses. Such eclipse observations of the corona (for visible wavelengths) were of the coronal "green" (5303 Å) and "red" (6374 Å) lines and it took many years to finally assess what was creating these strong emission lines[5]. Eminent authors at the time suggested that they were signatures formed by a new element called "coronium", but Mendeleev was near completion of the periodic table of elements and there was no space left for coronium, so something else had to explain these emission lines. Eventually, through the work on the spectra of highly ionised atoms of Edlén (1941) who, using the new models of atomic structure postulated by Grotrian (1928), identified the "red" line as belonging to nine times ionised Iron (Fe X). The follow up work (Edlén 1943) revealed that the corona was about one hundred times hotter than the photosphere.

This early work provided a clue to understanding the solar spectrum of emitted e-m radiation because it allowed the identification of specific lines with those belonging to particular atomic spectra observed in the laboratory plasmas of the time. However, these studies were restricted to optical wavelengths and to study fully the parts of the solar spectrum produced by hotter regions would require observations beyond the violet end of the optical spectrum (with the additional benefit of not being influenced by the intense photospheric radiation in that range). As we will see below, from ground based observatories, it is virtually impossible to make such observations because our atmosphere doesn't transmit radiation of wavelengths belonging to most emission lines formed in the transition region and corona (mostly in the ultraviolet, UV, or EUV, the extreme-UV and X-Rays; see Table 1.1) readily and we are required to take special steps to overcome this difficulty.

From pictures like those shown in figure 1.4 we can clearly observe the presence and influence of magnetic structures pervading the solar atmosphere. The vast number of possible morphological and topological magnetic fields generated by sub-photospheric dynamic convection mean that a course in solar zoology (or philately) is required to keep track of the newest "breeds". Possibly the most obvious determination to make by eye is the difference between the "quiet" and "active" Sun. Quiet Sun regions are simply identified by the fact that there appears to be very few or no complex magnetic structures present whereas ac-

[4] Visible - in the sense of naked eye observation.

[5] We now know that the bulk of coronal emission comes from the so-called "K-corona" (arising from electron scattering of photospheric light) whereas the portion attributed to the red and green lines is known as the "L-corona".

tive regions are usually associated with the most well known of solar features, sunspots and loop-like structures.

There are many other solar phenomena, both static and highly dynamic, associated with the interaction between the plasma and the magnetic fields present. To give a complete list of these, or at least those presently identified, would be inappropriate in this discussion but here are some of the most commonly observed: flares, surges, jets, polar plumes, prominences and coronal mass ejections (CMEs). Further discussion of these features is beyond the scope of this thesis, the interested reader is directed to the excellent texts mentioned above. The majority of the work we will present in due course will be most reliably applicable to steady quiet regions although modifications can (in some cases) be made to incorporate the timescales and dynamics of active regions.

1.2 Remote sensing of the Sun (1945 → Present)

We have mentioned above, in passing, that one of the major reasons why we have to place observing instruments above the Earth's atmosphere to study that of the Sun and of other more distant objects. There are three principal reasons why we should want to observed the Sun from space :

1. *Extending the range of wavelengths observable.* The hotter regions of the solar atmosphere emit in the far ultraviolet and X-ray spectral bands but our atmosphere is effectively

Table 1.1: The photon wavelength and energies of the electromagnetic spectrum.

Name	Wavelength range (λ Å)	Energy range (E eV)
Radio	$\geq 10^7$ Å	E $\leq$ 0.00124 eV
Infrared (IR)	$10^6 > \lambda \geq 7500$ Å	1.65 $\leq$ E $>$ 0.00124 eV
Visible	$7500 > \lambda \geq 3000$ Å	4.13 $\leq$ E $>$ 1.65 eV
Ultraviolet (UV)	$3000 > \lambda \geq 1500$ Å	8.24 $\leq$ E $>$ 4.13 eV
Extreme-UV (EUV)	$1500 > \lambda \geq 100$ Å	124 $\leq$ E $>$ 8.24 eV
Soft X-Ray (SXR)	$100 > \lambda \geq 1$ Å	12.4 $\leq$ E $>$ 0.124 keV
Hard X-Ray (HXR)	$1 > \lambda \geq 0.025$ Å	500 $\leq$ E $>$ 12.4 keV
Gamma Ray	$0.025 > \lambda$ Å	E $>$ 500 keV

opaque to that range of wavelengths. The only e-m radiation from the Sun that reaches Earth's surface is in the visible, a few "windows" in the near infrared, extremely high energy gamma rays and a wide range of radio wavelengths. Figure 1.2 shows the height of unit optical depth of the Earth's atmosphere as a function of wavelength.

2. *Reduction of scattering and distortion.* At visible wavelengths the corona is very faint compared to the extremely bright solar disc. Again, our atmosphere scatters light (cf. the blue appearance of the daytime sky is caused by the Rayleigh scattering of solar white light) and puts fundamental limits on distinguishing faint objects near bright ones. The distortion of light passing through the turbulent regions of our atmosphere (e.g. the troposphere) is another prime concern, but can be accommodated for by implementing complex adaptive optics schemes (see, e.g., Lloyd-Hart et al. 1998).

3. *Continuous observations.* For many long duration events such as monitoring oscillations of the photosphere and stochastic events like flares continuous observation is required just because of their particular physical nature. We can obtain continuous observations of the Sun in two ways :

   ⋆ placing a satellite in a Sun-synchronous orbit (a low Earth orbit running from pole to pole but in a slowly precessing plane which remains perpendicular to the Earth-Sun line).

   ⋆ placing a satellite into orbit at the Earth-Sun Lagrange point (along the Earth-Sun line at the point where the opposing gravitational attractions of the Sun and Earth cancel).

However, long before we had the advanced technology of today and were able to place an array of Sun observing satellites in orbit there were many successful attempts at remotely sensing the solar atmosphere. During World War II (WWII) solar physics was essentially a classified subject and the research was for military application. However, soon after the close of hostilities, UV observations of the Sun were made using a spectrograph flown on a slightly modified versions of Werner Von Braun's infamous V-2 rockets. The capture of several V-2 rockets, the repatriation of various engineers and technicians, and the resulting technological advances of the late 1940's allowed the Sun to be studied regularly in the UV and X-Ray wavelength bands. These observations were made using 'sounding rockets'[6] which are still

[6] 'Sounding' comes from a nautical term for taking a measurement by dropping a line into the sea.

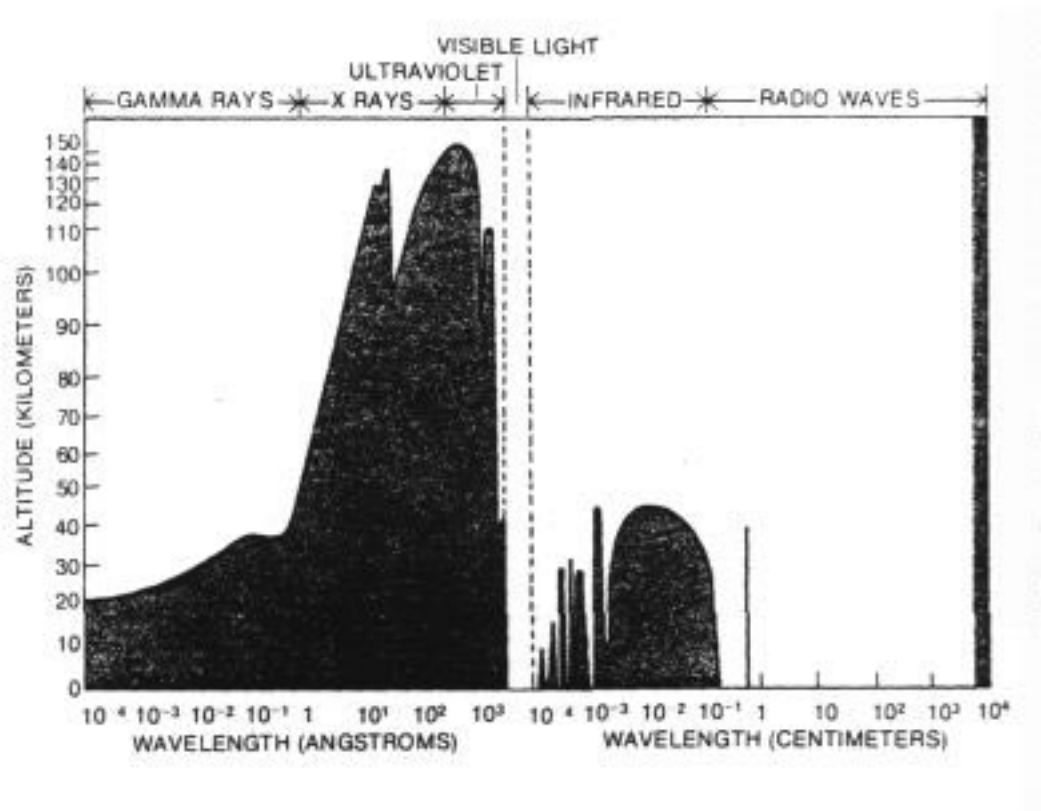

Figure 1.2: Plot showing the height of unit optical depth of the Earth's atmosphere as a function of wavelength. Clearly visible are the wavelength ranges where the Earth's atmosphere is effectively opaque (taken from Golub & Pasachoff 1997).

frequently used to supplement and calibrate observation made by Earth orbiting satellites.

In the late 1950s, virtually as soon as the first earth orbiting satellites were being successfully launched, the theoretical development of space-borne observations took great leaps with instrumentation placed on the *Sputnik* series. Soon after the turn of the decade (1962) NASA launched the first of the Orbiting Solar Observatories (OSOs) which spearheaded their research objectives until the middle of the 1970s with the launch of OSO-8. The OSOs were designed to cover the UV and extreme-UV regions and also showed up a flaw in observations made by earlier orbiting satellites; the effect of background radiation from electrons in the Earth's Van Allen radiation belts caused severe contamination of the measurements. In the mean time NASA had developed, built and placed in orbit (using materials from the Apollo moon programme), the *Skylab* space station which was operational for a period of 251 days from May 1973 until its eventual re-entry in July 1979.

Skylab saw the introduction of imaging technology that had higher spectral and spatial resolution than any of its predecessors. The hub of the Skylab observations was the Apollo Telescope Mount (ATM), the home for eight full size instruments that covered the entire spectral region from 2 Å to 7000 Å. The Skylab mission, because of increased observation time and scientific funding levels, was the most productive mission of solar observations from space. The advances of the mission are documented in the proceedings of three workshops (Zirker 1977; Sturrock 1980; Orrall 1981) on the evolution and structure of "coronal holes", active regions and the analysis, observation and predictions of solar flares.

In the years following its demise the quantity of high quality Skylab data kept solar physicists busy, that is until the mid to late 1980s with the launch of several new missions (e.g., *Hinotori* and the *Compton Gamma Ray Observatory*) but in particular the NASA/NASDA YohKoh ("sunbeam" in Japanese) in 1991. YohKoh is a joint mission that is investigating the solar corona and its X-Ray emission. Its principal aim was to obtain data of unprecedented quality on the emissions from active regions and flares up to and through the last solar maximum in 1992. YohKoh is still "going strong" and providing vast quantities of information rich data for the community as the solar activity cycle is on the rise again.

This brings us up to date, as far as solar observing missions are concerned, apart from the mission for which this thesis is intended as a theoretical aid to compliment and enhance the physical understanding of the observations made. The mission to which we refer is the ESA/NASA "cornerstone 2000" satellite called the Solar and Heliospheric Observatory, or SOHO for short.

1.2.1 The Solar and Heliospheric Observatory (1995 → Present)

The Solar and Heliospheric Observatory (SOHO) was launched into orbit from Cape Canaveral at 8:08 UT on December 2nd 1995. It was placed in a halo orbit around the L1 Earth-Sun Lagrange point some 1.6 million kilometers from Earth and it took nearly four months to get there. Figure 1.3 shows a schematic layout of the SOHO spacecraft which contains the largest compliment of solar observation tools since Skylab was constructed some twenty years previously.

SOHO's prime scientific goals are laid out in detail in Fleck et al. (1995) and these include understanding :

⋆ the structure, composition and dynamics in the solar interior (the region below $\tau_{5000} = 1$, i.e. below the photosphere)

⋆ the structure and dynamics of the chromosphere, transition region and corona

⋆ the solar wind and its interaction with the Earth's atmosphere

The second of these objectives is our principal concern; diagnosis and interpretation of the emission of the outer regions of the solar atmosphere. Images such as those in figure 1.4 show the different features visible in observations at different wavelengths that are taken almost simultaneously. Table 1.2 gives some details of the instruments used for this study and certain other attributes. From this set of telescopes and spectrometers we obtain plasma diagnostics which provide us with temperature, density and velocity measurements of the emitting material. In particular though, we discuss the theoretical development for data acquired by the Coronal Diagnostic Spectrometer (CDS; Harrison et al. 1995) and the Solar Ultraviolet Measurement of Emitted Radiation (SUMER; Wilhelm et al. 1995) instruments.

1.3 The structure of this thesis in brief

We have seen that probing the outer atmosphere of the Sun requires a great deal of planning and accurate execution to place intricate pieces of unmanned machinery over a million kilometers into space. Our discussion so far has raised one very important question, how do we explain the heating of the chromosphere and corona in particular ?

We will **not** attempt to answer this question but we hope to achieve the next best thing with this thesis. That is, we will endeavour to enhance the understanding, methodology

Table 1.2: Brief details of the spectroscopic instruments on SOHO for remotely sensing the solar atmosphere. Note that NIS and GIS are the Normal Incidence and Grazing Incidence Spectrometers that constitute CDS.

Acronym/Investigation		Wavelength Range (Å)
SUMER	1st Order	390-805 Å
Solar UV Measurement of Emitted Radiation	2nd Order	780-1610 Å
CDS	NIS	308-381, 513-633 Å
Coronal Diagnostic Spectrometer	GIS	151-221, 256-338, 393-493, 656-785 Å
EIT	4 filters	171 Å (Fe X/IX), 195 Å (Fe XII),
Extreme UV Imaging Telescope		284 Å (Fe XV), 304 Å (He II)
UVCS	3 channels	1145-1287 Å (Ly-α), $\sim$1032 Å (O VI),
UV Coronagraph Spectrometer		4500-6000 Å (White Light Channel; WLC)

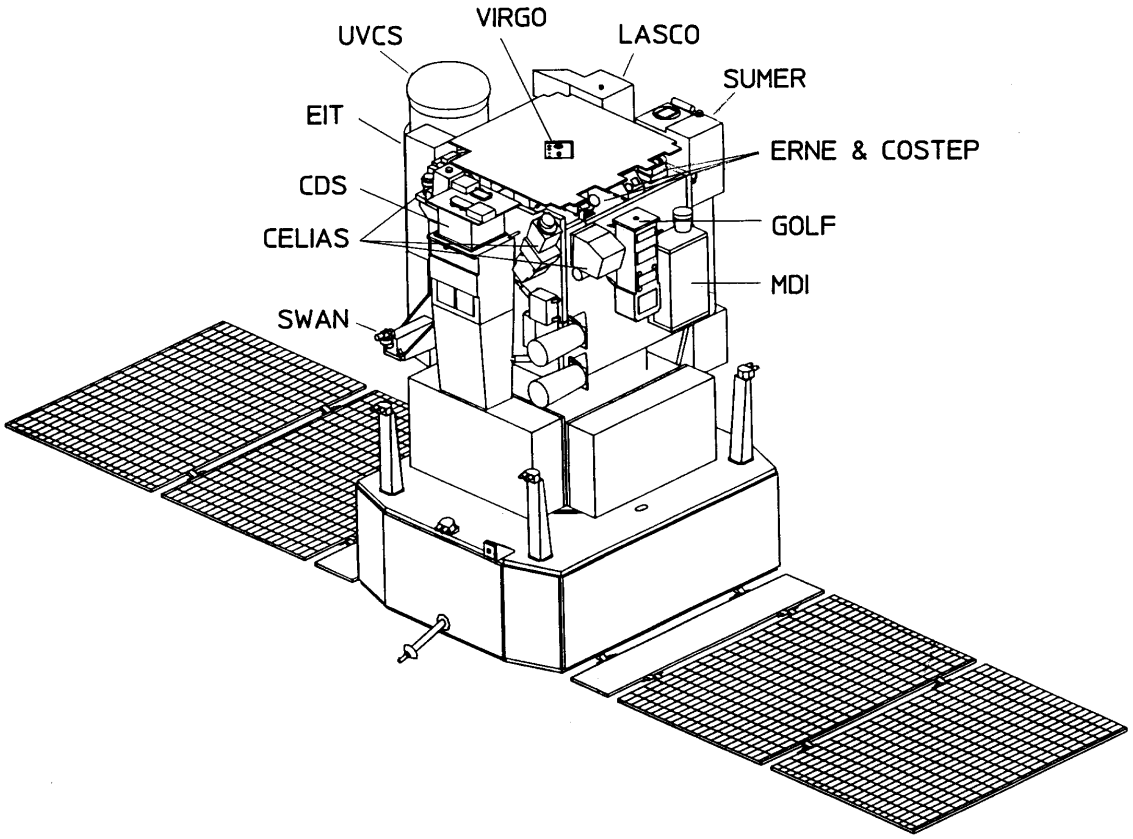

Figure 1.3: Schematic of the SOHO satellite and its payload of scientific instruments. Some details of the spectroscopic instruments can be found in Table 1.2 (taken from Fleck et al. 1995).

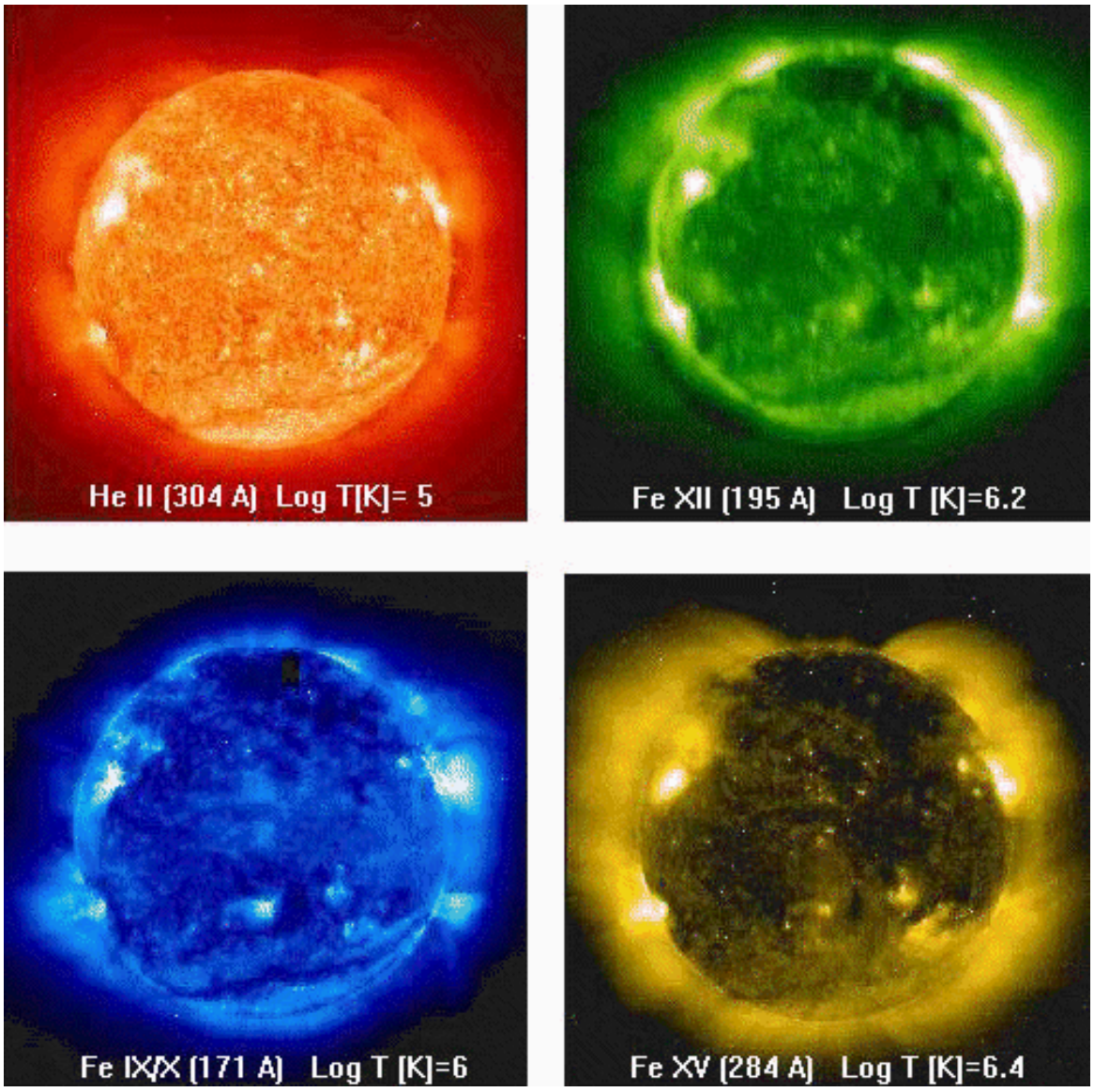

Figure 1.4: Images taken by the Extreme-ultraviolet Imaging Telescope (EIT) on June 24$^{\text{th}}$ 1998 detailing some of the solar structures seen in quiet and active regions of the Sun. Each image is taken in one of EIT's four bandpasses each of which images the EUV emission of lines formed at different temperatures and hence solar altitudes (see, e.g., figure 1.1). These images cover temperatures from around 50,000 K (He II) in the transition region to 2.5 million K (Fe XV) in the corona. Interestingly the upper right image appears out of focus, indeed this image appears to be the last taken by EIT at 23:19 on the date above before the satellite was temporarily lost.

and implementation employed when seeking diagnostic quantities from the observed solar UV/EUV spectra. Only once these can be obtained with a high degree of uniqueness can we proceed to infer the likely form of physical processes occuring in the emitting plasma.

We have already mentioned the ground-breaking work of Edlén (1941) during WWII that was rapidly followed by the work of Menzel et al. (1941) who made use of these theoretical advances to make estimates of the electron densities and physical structure of planetary nebulae from emission line ratios. Today, much of the analysis presented on plasma diagnostics follows roughly the same theme however the work of Jefferies et al. (1972a, b) gave it a slightly different slant. They re-cast what was essentially the work of Pottasch (1964) for obtaining electron density and temperature distributions, Differential Emission Measures or DEMs, into an "inverse problem". Stated basically, this means that the underlying plasma "source" (DEM) is convolved through the atomic mechanisms to produce the observed emission spectrum and the object of the problem is that; given the observed spectrum and the theoretical atomic models (both known) what is the nature and structure of the source (unknown) ? They set out to formulate this problem, but offered no formal solution. Advances in the formulation and understanding of the nature of this inverse problem were made by Craig & Brown (1976), Almleaky et al. (1989) and Brown et al. (1991) and we will meet these in due course, but the most recent publication (treating the most general case of density and temperature distributions) of Judge et al. (1997) has shed light and posed serious questions about the limitations of reliably recovering "useful" diagnostics. Their paper, titled "Fundamental limitations of emission-line spectra as diagnostics of plasma temperature and density structure", considers in detail, the effect on the inversion of uncertainties in the (hitherto assumed known) atomic calculations. Much of the work presented in this thesis is motivated by the points raised by Judge et al. (1997) and we show that novel methods can be employed to minimise some of the difficulties they encountered.

In the following chapter we introduce the necessary terminology and methodology to understand and obtain solutions to inverse problems (Section 2.1) and to construct relevant atomic models and discuss possible sources of uncertainty therein (Section 2.2).

Chapter 3 digresses a from the main theme and flow to introduce a valuable diagnostic tool (used extensively in the following chapters) called a Genetic Algorithm (GA; Goldberg 1989). In particular, we concentrate on the application of GAs to the decomposition of emission line spectra. The method discussed therein is applied to several synthetic spectra and a UV line emission spectrum taken by the aforementioned SUMER instrument. We

show that these GA based decompositions are stable to data noise and to the effects inherent to poorly sampled spectra, especially at the limits of instrumental resolution. The material contained in Chapter 3 was published as McIntosh et al. (1998b).

Chapter 4 sees a return to the main theme of this thesis, the inference of reliable plasma diagnostics from emission line spectra in the wavelength range of the SOHO CDS/SUMER instruments. We, in Section 4.1, approach this from two different perspectives, the line ratio and the DEM methods, however we show that both are mathematically equivalent (Section 4.1 was published as McIntosh et al. 1998a). We will show that the line ratio method is an adequate means of overcoming the theoretical uncertainties discussed in Judge et al. (1997) and that formal inversion to obtain the aforementioned DEMs is the *only* way, in the context of inverse methods (including those related methods using line ratios), to extract useful information from emission line spectra (Craig & Brown 1976). To this end, we (in Section 4.2) present a novel GA based approach, the Ratio Inversion Technique (RIT), to 'couple' the two methods mentioned above and obtain the most reliable possible DEMs in the presence of these theoretical uncertainties. We show that the RIT exploits the systematic nature of these uncertainties and obtains DEMs to a higher degree of uniqueness than a standard DEM inversion in their presence. Section 4.4 sees the application of the RIT to emission line spectra obtained by the Solar EUV Rocket Telescope and Spectrograph (SERTS--89; Thomas & Neupert 1994) to see the differences occurring between DEMs obtained by RIT inversion and those published previously (Brickhouse et al. 1995; Landi & Landini 1997; Lanzafame et al. 1998).

In Chapter 5 we employ another GA based method (SELECTOR) to overcome another serious problem associated to the solution of any inverse problem, in this case we focus on the univariate (T_e and n_e) DEM inverse problems. This discussion concerns the amplification of errors in the observed emission line intensities to catastrophic errors in the recovered DEMs through the linear dependence (amongst other things) of the inverse operator, or *kernel*, which is then known as being *poorly conditioned* (see Craig & Brown 1986). The method we present searches a list of 133 emission lines (again in the CDS/SUMER wavelength range) to identify the subset of those lines that reduces the error amplification by obtaining the kernel with the best possible conditioning.

Chapter 6 discusses the points raised and methods presented in the preceding chapters as well suggesting possible applications, improvements and future extensions. In all we show that careful consideration of the physical nature of the emission lines used as well as careful

selection can yield results that are numerically stable and offer a greater degree of uniqueness than those obtained using a standard approach.

Chapter 2

An introduction to inverse problems and plasma diagnostics

This Chapter

In this chapter we discuss the essential theoretical and analytic methods employed in the following chapters of this thesis. We concentrate upon the formulation and numerical solution of inverse problems, presenting various practical 'tools' en route. Further, we discuss the theoretical foundations of obtaining reliable diagnostics of hot solar plasmas and likely sources of uncertainty therein.

2.1 Inverse Problems

Inverse problems occur in a wide variety of physical contexts. They are a natural consequence of any situation in which an observer makes an indirect measurement of the quantities which he or she is actually interested in. Indeed, the designation 'inverse' arises from the fact that many objects of interest are manifestly obscured from observation, either by their physical location (such as the solar corona), or their intrinsic non-measurability (e.g., the sub-surface velocity of volcanic magma, cannot be measured directly since any measuring probe would likely be destroyed by the enormous stresses and temperature). This situation essentially defines "remote-sensing" sciences such as astronomy where the observed quantities are obtained from the electromagnetic radiation emitted by atoms, ions and electrons interacting under external forces. Very often the source is not resolved spatially, so only the volume integrated particle/photon flux from the object can be measured. Theoretical considerations

must then provide a relationship between the sets of observables (often of secondary interest) and non-observables (unknowns) of primary interest. These are often related in a non-trivial way (i.e. they are *coupled*). Obviously, without theoretical modelling of such coupled systems it would be impossible to learn anything at all about such physical sources. In many cases this coupling, or convolution, gives rise to the well known (even if not well understood) difficulties with solving inverse problems : namely *ill-posedness* and *poor conditioning*. Indeed, such difficulties manifest themselves by creating non-uniqueness and instability respectively of the solution, even from small perturbations in the observed quantities.

To be mathematically concise, inverse problems are a special class of functional equations encompassing all classes of integral, differential and matrix equations. In this thesis we will only consider inverse problems relating to the solution (which will be, more often than not, numerical) of integral equations; often requiring specific mathematical techniques to achieve a numerically stable solution, i.e. counteracting the difficulties mentioned above. In general, when attempting to find a stable solution to an inverse problem we will make use of some prior physical knowledge or assumptions about the nature of the problem. Before we consider a specific example of an inverse problem in solar physics we extract a little of the preface to Craig & Brown (1986) as a very apt description of inverse problems in an astronomical context

> "The remote observer finds himself in a situation, akin to that of a spectator at a magic show, where he is presented with a limited set of more or less remarkable data emanating from a source, the nature of which he is fascinated to discover but which he is not permitted to handle directly. In the magic show, the basic mechanism of the trick known only to the magician, is convoluted through the unrevealed process of his presentation, before appearing in strongly modified form to the spectator. In astronomy, the unknown basic physics of the observed source is convoluted through the source structure and emission processes (also unknown) before arriving at the observer's instrument."

A physical example of such a process lies in the photon spectrum of a solar flare. The observed photon spectrum can be regarded as a convoluted representation of the electron spectrum. To obtain the energy distribution of electrons in the flare, we use physical information (and assumptions) about electron collision processes (cross-sections, etc) and the electron distribution to cast the form of the convolution *operator* and to stabilise the inversion respectively.

Such information allows us to "step back" from the observed photon spectrum to the inferred electron spectrum. This *a priori* information provides a means of obtaining the most 'reliable' solution when the operators present are such as to make the inferred solution *extremely* sensitive to errors in the observed quantities or are very nearly *singular* (i.e. a linear operator $\mathcal{S}$ is singular precisely when it has *no* corresponding inverse operator $\mathcal{S}^{-1}$).

This section provides the necessary mathematical framework needed to solve inverse problems numerically. There are many mathematical and numerical techniques to help obtain a *stable* solution of inverse problems. We seek the ones which will help us to "make the most of what we have got", i.e. obtain as much information about the physical source from the limited amount of available observed data. Section 2.1.1 details the two different classes of integral equations and how they can be cast as matrix equations whereas Section 2.1.2 describes the effects of ill-posedness and poor conditioning of inverse problems mentioned earlier. Numerical solution techniques are described in Section 2.1.3, particularly the methods known as Singular Value Decomposition (SVD), Quadratic Regularisation (QR) and Maximum Entropy (ME). These algorithms and techniques take great advantage, as we shall see, of the relationship between *linear* inverse problems and linear systems of equations (discussed at greater length in Craig & Brown (1986) and references therein) thus making analysis relatively straightforward for such a mathematical abstraction.

2.1.1 Mathematical definitions

In order to formulate a mathematical description of an inverse problem one must establish a relationship between the observable quantities $\mathbf{y}$ of a particular problem, and the set of non-observables, $\mathbf{x}$. In general $\mathbf{y}$ and $\mathbf{x}$ are symbols that describe a number of pieces of information. If we consider only the case where the number of observable quantities and unknowns are finite, we can write $\mathbf{y} \equiv \{y_i; i = 1, \ldots, n\}$ and $\mathbf{x} \equiv \{x_j; j = 1, \ldots, m\}$ for some positive integers n and m. However, it is possible that the observables or the unknowns (or both) are values of functions of a continuous (real) variable so that there is an infinite number of pieces of information (conceptually, functions of a continuous variable may be considered as 'vectors' in an infinite dimensional vector space). In practice, the observables are the remotely sensed data of the problem. For example, the emission line intensities in the Differential Emission Measure problem of later chapters, or the measured frequency splittings in Helioseismology differential rotation inverse problems (see, e.g., Hansen 1994) and the unknowns are the potentially continuous "source" functions.

Any relationship derived from a mathematical model of a physical process can be written, without loss of generality, as the relationship between $\mathbf{y}$ and $\mathbf{x}$

$$\mathbf{G}(\mathbf{y}) = \mathbf{K}(\mathbf{x}) \quad , \tag{2.1}$$

where $\mathbf{G}$ and $\mathbf{K}$ represent some *known* functions of the observables and non-observables, respectively. The term 'function' is used in its broader mathematical sense : $\mathbf{K}$ and $\mathbf{G}$ are mappings from the (vector or function) spaces containing the aforementioned quantities. Equality in equation (2.1) forces $\mathbf{G}(\mathbf{y})$ and $\mathbf{K}(\mathbf{x})$ to have the *same* number of degrees of freedom and to form a system of equations, whilst its classification as an inverse problem depends entirely on the properties of $\mathbf{K}$ and holds when $\mathbf{K}$ is a non-trivial function of the non-observable $\mathbf{x}$ (i.e. the system of equations is coupled as previously noted).

Although equation (2.1) is general in nature it can be used to categorise inverse problems. Consider the following specific examples :

1. Suppose $\mathbf{y}$ and $\mathbf{x}$ are vectors, of dimension m and n, respectively (m data values and n unknown parameters to find). Then $\mathbf{G}(\mathbf{y})$, and therefore $\mathbf{K}(\mathbf{x})$, is a vector, of length q, say. Equation (2.1) becomes, in terms of vector components,

$$G_i(\mathbf{y}) = K_i(\mathbf{x}), \text{ for } i = 1, \ldots, q. \tag{2.2}$$

   If $\mathbf{G}$ and $\mathbf{K}$ are both linear functions, then equation (2.1) may be written, using matrix notation as, $G\mathbf{y} = K\mathbf{x}$, where G and K are $q \times m$ and $q \times n$ matrices, respectively. Indeed for $q = m$ and $G = I_m$ (where I_m is the identity matrix of dimension m) equation (2.1) becomes a pure matrix-type inverse problem of the form, $\mathbf{y} = M\mathbf{x}$.

2. Now we suppose that $\mathbf{y}$ is a vector as before, but $\mathbf{x}$ is a function ($\mathbf{x} = x(t)$) of some real variable t ($u \leq t \leq v$). For simplicity we assume that $\mathbf{K}$ a linear function of $\mathbf{x}$ so that, if $\mathbf{G}(\mathbf{y})$ is a function of s (again a real variable with $a \leq s \leq b$), equation (2.1) becomes (noting that when $\mathbf{x}$ represents the values of a function of a continuous variable, t in this case, $\mathbf{K}$ involves an integral over that variable)

$$G(\mathbf{y}; s) = \int_u^v k(s;t)x(t)dt, \text{ for } a \leq s \leq b. \tag{2.3}$$

   This is the general form of a *Fredholm integral equation* (see Craig & Brown 1986) and is also discussed at greater length in Section 2.1.1.1.

A more appropriate treatment of $\mathbf{G}$ as the *data* of the problem is to consider the replacement of $\mathbf{G}(\mathbf{y})$, in equation (2.1), with the symbol $\mathbf{g}$, so that we study problems that take the

form

$$\mathbf{g} = \mathbf{K}(\mathbf{x}). \tag{2.4}$$

On inspection of equations (2.2) and (2.3), we see how the integral equation is directly analogous to the linear matrix system of equation (2.4) when $\mathbf{K}$ is a linear functional, as before. Indeed, many of the properties discussed in this, and other chapters depend on standard linear matrix operations. These suffer from various "defects". One principal defect in many of the linear matrix adaptations of inverse problems is that of *singularity*. In terms of matrices, singularity is a familiar phenomenon. Consider the solution of a square matrix equation $\mathbf{y} = A\mathbf{x}$ (for vectors $\mathbf{x}$ and $\mathbf{y}$ and square matrix A) which is obviously $\mathbf{x} = A^{-1}\mathbf{y}$ provided that A^{-1} exists. This is precisely when it has *no* linear dependence in its rows thus it has non-zero determinant and is classed as *non-singular*. However, when treating matrices derived from integral operators (cf. $\mathbf{K}$ above) singularity is dependent on its eigenvalues with the degree of singularity given by the number of *zero*[1] eigenvalues and again the number of zero eigenvalues directly indicates the degree of linear dependence in the kernel operator. The concept of singularity is discussed further in Section 2.1.3.2.

2.1.1.1 Fredholm integral equations

By far the most general class of integral equation is the Fredholm integral equation (cf. equation (2.3)). Note that

$$\int_a^b k(x,y)f(x)dx = g(y) \quad c \leq y \leq d \text{ and} \tag{2.5}$$

$$f(y) + \lambda \int_a^b k(x,y)f(x)dx = g(y) \quad c \leq y \leq d \tag{2.6}$$

are Fredholm integral equations of the first and second kind respectively.

The analytical solution to equation (2.5) (and equation (2.6)) is the continuous function $f(x)$. However in the 'real world' there are only a finite number of observables available, and not the infinite number required for the exact recovery of $f(x)$ from either equation if we neglect, for the moment at least, ill-posedness and poor conditioning. Thus, we must solve the integral equation over a discrete set of values. To discretise the integral equation we use linear quadrature methods (cf. the Trapezoidal rule or Simpson's method discussed in Press et al. 1992) to 'break up' the integrand and integrate it over a shorter range than a to b

[1] In this sense, "zero" can be interpreted as numerically zero, at the precision of the computer used.

forming an n element data vector $\mathbf{g}$ with each element $g(y_i)$ given by

$$g(y_i) \;=\; g_i = \sum_{j=1}^{n} \int_{x_{j-1}}^{x_j} k(x, y_i) f(x) dx \;, \tag{2.7}$$

provided that $x_1 = a$ and $x_n = b$. If then we assume that the 'mesh' size (i.e. $\frac{|b-a|}{n}$) is fine enough for $f(x)$ to be constant 'enough' over that interval to be approximated by its value at the midpoint, then we have

$$g_i \;=\; \left(\sum_{j=1}^{n} \int_{x_{j-1}}^{x_j} k(x, y_i) dx \right) \; f(x_j) \;. \tag{2.8}$$

Equation (2.8) is analogous to the equation of a vector element g_i given by

$$g_i \;=\; \sum_{j=1}^{n} K_{ij} \; f_j \;, \tag{2.9}$$

where we have equated $\int_{x_{j-1}}^{x_j} k(x, y_i) \; dx$ and $f(x_j)$ with the matrix element K_{ij} and vector element f_j respectively. So, to generalise, for all data elements $\mathbf{g}$ we have the matrix equation

$$\mathbf{g} \;=\; K \; \mathbf{f} \;, \tag{2.10}$$

and the solution of Fredholm integral equations of the first type reduces to solving linear matrix equations like those mentioned immediately above.

When the physical model requires that the source function is the solution of a Fredholm equation of the second kind (see, e.g., equation (2.6)) the integral form discretises to a form similar to the classical eigenvalue problem

$$\mathbf{g} \;=\; (I - \lambda K) \; \mathbf{f} \;, \tag{2.11}$$

where I is the identity matrix of dimension $n \times n$. However solution of the discrete form of equation (2.6) is not trivial (Bertero 1997) and since such a treatment is outwith the scope of this work will be considered no further.

To digress a little at this point, a Singular Value Decomposition (SVD; see Section 4.3.2 of Craig & Brown 1986) of the matrix K in equation (2.10) can yield very important properties of the inverse problem as well as allowing the extension of inverse methodology from square to non-square matrix equations. Given that K is a $m \times n$ matrix ($m,\; n$ integers not necessarily the same), on performing the SVD we obtain

$$K \;=\; U \; \Sigma \; V^T \;, \tag{2.12}$$

where the matrices U $(m \times n)$, V $(n \times m)$ and Σ $(m \times m)$ are orthogonal. The columns of U, and V are the *singular vectors* $\mathbf{u}$, and $\mathbf{v}$, respectively and Σ is a diagonal matrix containing

the *singular values* $(\sigma_1, \ldots, \sigma_n)$. In short, the singular vectors of K are the non-zero vectors satisfying both

$$K\ \mathbf{v}_i = \sigma_i\ \mathbf{u}_i \text{ and} \tag{2.13}$$

$$K^T\ \mathbf{u}_i = \sigma_i\ \mathbf{v}_i \tag{2.14}$$

with $K\ K^T\ \mathbf{u}_i = \sigma_i^2\ \mathbf{u}_i$ and $K^T\ K\ \mathbf{v}_i = \sigma_i^2\ \mathbf{v}_i$ also holding such that the vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are the eigenvectors of symmetric matrices $K\ K^T$ and $K^T\ K$ respectively with corresponding eigenvalue σ_i^2. These left $(\mathbf{u}_i)$ and right $(\mathbf{v}_i)$ singular vectors form orthonormal bases for K, i.e. the orthogonal matrices U and V in equation (2.12). This relationship allows us to diagnose how poorly conditioned the inverse problem is, as discussed in the next section.

2.1.1.2 An example of a Fredholm equation : The Differential Emission Measure problem

This thesis contains a detailed discussion of one particular inverse problem that takes the form of a Fredholm equation of the first kind. The aim of this thesis is to 'reliably' infer the solar plasma source distribution in terms of a Differential Emission Measure (DEM) function from remotely sensed line intensities from Ultraviolet (UV) and extreme-Ultraviolet (EUV) emission line spectra (see Section 2.2.1.1). From equation (2.73) we see that the integrated line intensity of an emission line, with identifier i, as a function of electron temperature (T_e) is

$$I_i = \int_{T_e} K_i(T_e)\xi(T_e)\ dT_e\ , \tag{2.15}$$

where $K_i(T_e)$ is the emissivity of the line and $\xi(T_e)$ is the temperature DEM function.

To perform the inversion, and infer $\xi(T_e)$, we have to observe a set of n $(n > 1)$ emission lines so that equation (2.15) takes on the matrix form of equation (2.10) by using each I_i and $K_i(T_e)$ as the i^{th} element of $\mathbf{g}$ and row of K respectively. The resulting kernel matrices are highly singular but we leave discussion of their singularity to Chapter 5. The DEM inverse problem is a *very* real case of an ill-posed inverse problem; it is common to see different authors using the same data, but producing very different inferred emitting plasma structures, see, e.g., Section 4.4 and Kashyap & Drake (1998).

2.1.1.3 Volterra integral equations

The class of integral equations known as Volterra equations may be regarded as a special case, or sub-class, of Fredholm equations when the kernels exhibit a "cut-off" or when the

variable appears in the limits of the integral. The difference between the formulation of the two arises because we retain the constant a but have made b some function of x ($b = b(x)$) in equations (2.5) and (2.6), so we have

$$\int_a^{b(x)} k(x,y)f(x)dx = g(y) \ . \tag{2.16}$$

In fact, Volterra equations (of the first kind) may be treated similarly to equation (2.5) with $k(x,y) = 0$ for $y > x$, this 'truncation' gives them a quite distinct nature. The associated kernel matrix of the discretised form is lower-triangular[2] and the linear system has a recursive nature easily amenable to the *Gaussian Elimination* or *back-substitution* methods discussed in Sneddon (1972). The solution to many inverse problems in the physical sciences reduce to the solution of such equations, e.g. see the following example.

2.1.1.4 An example of a Volterra equation : Non-thermal bremsstrahlung spectra

As mentioned above, if we wish to infer the average[3] source electron energy spectrum, $F(E)$, of a beam or flare it can only be done from the properties of the radiation such as the observed bremsstrahlung spatially integrated photon spectrum, $J(\epsilon)$, of the source. As described in Brown (1971) (and Craig & Brown 1986) we have the form

$$J(\epsilon) = \bar{n}_p V \int_\epsilon^\infty Q_B(\epsilon, E)\bar{F}(E)\ dE \tag{2.17}$$

where $\bar{n}_p = \frac{1}{V}\int_V n_p(\mathbf{r})dV$ and $\bar{F}(E) = \frac{1}{\bar{n}_p}\int_V F(E,\mathbf{r})n_p(\mathbf{r})dV$, V is the source volume and n_p is the proton density and $Q_B(\epsilon, E)$ is the electron-ion (non-relativistic) Bethe-Heitler bremsstrahlung cross-section (see Brown 1971, 1978).

Equation (2.17) is of Volterra type as can be more clearly observed by changing variable from E to $y = 1/E$ and $x = 1/\epsilon$

$$g(x) = \int_0^x \frac{f(y)dy}{(x-y)^{1/2}} \tag{2.18}$$

where $f(y) = y^{-2}\bar{F}(1/y)$ and $g(x) = x^{-1/2}G(J(1/x))$. This problem, which is moderately ill-posed, has been widely researched (see, e.g., Brown et al. 1998) and gives rise to electron energy spectra which are unstable to noise in $g(x)$, and in more general cases non-unique (depending on the physical characteristics of the model atmosphere used, i.e. full or partial ionisation).

[2] *All* kernels of Volterra equations can be expressed as Heaviside functions in K.

[3] Average in the sense that $F(E,\mathbf{r})$ it is a function of three spatial coordinates.

2.1.2 The ill-posed inverse problem

By far the biggest source of anxiety when attempting to find *the* solution of an inverse problem is of a philosophical nature. Consider the scientist who, on testing a new hypothesis, calculates (using the new theoretical model) a set of synthetic 'observed' quantities. From these the scientists hopes to reconstruct the generating source function precisely. The scientist typically discovers that from his *one* data set many (some physically unsuitable) source functions may appropriately "fit the bill" (see, e.g., figure 2.1). Such failure is common and seemingly unnoticed in many areas of the physical sciences, but is often easily cured by proper consideration of the solution space and prior restrictions on the properties on the solution. Here are just three of the things to be considered when solving an inverse problem of *any* kind :

1. The ill-posedness of the integral equation itself, i.e. the non-uniqueness of the recovered solution. This depends critically on the integral operator (kernel) of the problem.

2. The amount by which errors in the data or observed quantities become amplified in the recovered solution is due to the *conditioning* of the kernel matrix. A very poorly conditioned kernel can lead to incredibly unphysical solutions and also to a multiplication of physical ones - all are 'acceptable' in the sense of fitting noisy data.

3. Indeed, the solution may, for noisy data, lie in an undesirable sub-space of the solution space, e.g. the recovered solution is negative at some point yet we used a strictly positive source model to construct the test data. Hence, it is usually essential to impose *a priori* constraints, such as smoothness, to our solution in order that it "makes sense".

All of these factors are discussed in the following text with enough detail to give the reader a grasp of the difficulties involved in solving ill-posed inverse problems.

In any vector or function space it is important to have an estimate of the magnitude of a quantity (e.g. $|\mathbf{v}|$ on the real line). Indeed, there are many fundamental properties of inverse problems that can only be described by knowledge of the *metric* or *norm* of the function or vector space they inhabit. The magnitude measure of an element of a particular space is called its *norm*, and if the spaces $\mathcal{S}$ and $\mathcal{D}$ are those containing the solution and data respectively, their norms are correspondingly denoted as $\|.\|_{\mathcal{S}}$ and $\|.\|_{\mathcal{D}}$. As we will see, norms are vital to making estimates of error amplification in the solution of integral equations and inverse problems alike.

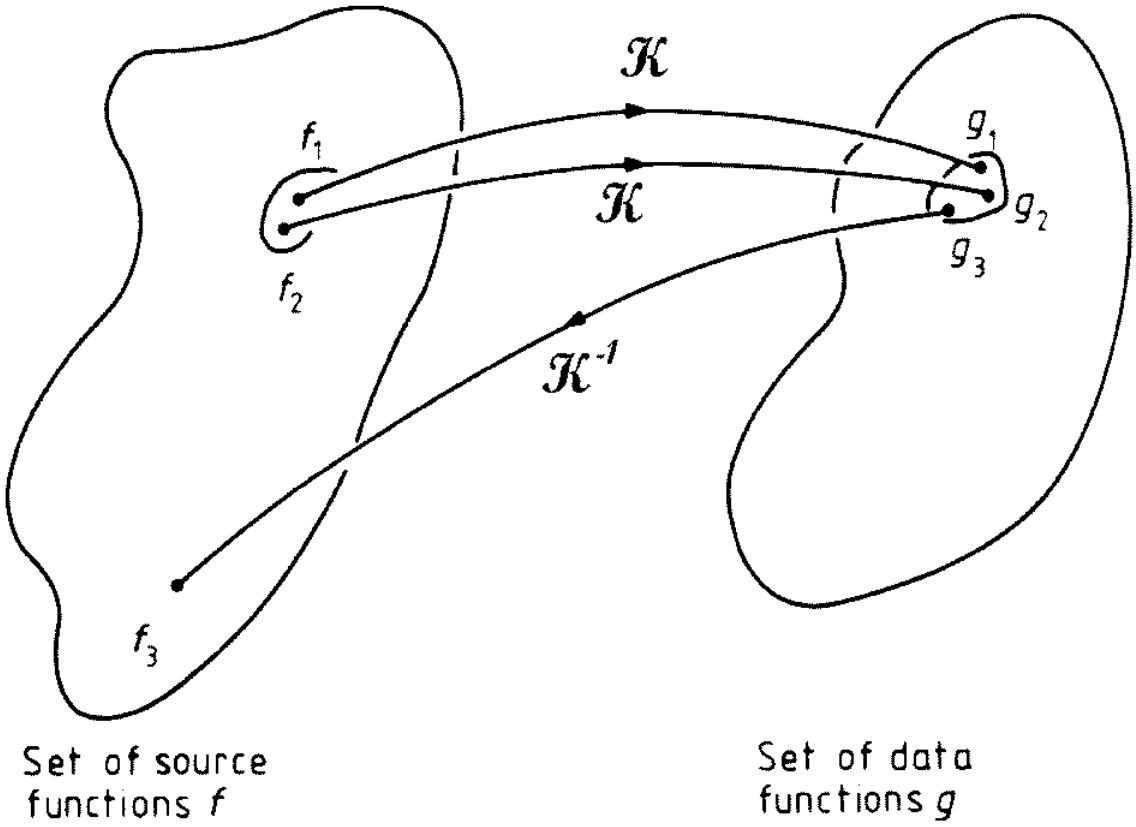

Figure 2.1: An example (taken from Craig & Brown 1986 pg., 47) showing that, in a function space, data functions g_i that are close ($\|g_i - g_j\|_{\mathcal{D}} < \varepsilon$ where ε is small) to the *exact* data function can produce, through the inverse mapping $f_3 = \mathcal{K}^{-1}\, g_3$, unphysical solutions that are arbitrarily far from the *true* solution.

As a temporary digression and for future reference, it is useful here to define the general properties of *norms* (or metrics) on vector spaces : A norm $\|.\|_{\mathcal{V}}$ on a vector space $\mathcal{V}$ is any function f ($f : \mathcal{V} \mapsto \mathbf{R}$, the space of real numbers) that satisfies the following three properties

1. $\|\mathbf{x}\| \geq 0$, and $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} \equiv \mathbf{0}$

2. $\|\alpha\mathbf{x}\| = |\alpha| \ \|\mathbf{x}\|$ for any real number α

3. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

for any vectors $\mathbf{x}$ and $\mathbf{y}$ in the space $\mathcal{V}$. Consider also the norm of a matrix $M_{m\times n}$ in the space $\mathcal{M}$ of real $m \times n$ matrices given by $\|M\|_i$ where i is the order of the norm. The most commonly used norms are the Euclidean distance or 2-norm

$$\|M\|_2 = \sigma_{max} \ , \tag{2.19}$$

where σ_{max} is the maximum singular value of K defined previously (see also Section 2.1.3.2). Similarly, the ∞-norm or the 'Infinity' metric as it is also known, is given by

$$\|M\|_\infty = \{\max_i \sum_{j=1}^{n} |M_{ij}|\} \tag{2.20}$$

and is the sum of the elements in the maximum row of M.

As stated above, the essence of an ill-posed problem is the instability introduced in the recovered solution; consider the Fredholm integral equation of the first kind given by

$$\int_0^\infty k(x,y)\ f(x)\ dx = g(y) \ , \tag{2.21}$$

which has the *regularised* (see Section 2.1.3.1) solution vector, $\hat{\mathbf{f}}$, discretised over a fixed mesh. Naively, the problem appears to be completely solved, but seldom is this the case. The cause of the non-uniqueness in the solution is that functions in the operator *null-space* are being linearly superimposed onto the actual solution. In the notation of the previous section we have the situation that, for any non-zero vector $\mathbf{x}$ in $\mathcal{S}$ there exist, in the data space $\mathcal{D}$, vectors $\mathbf{g}(\mathbf{y}) = K(\mathbf{x})$ that satisfy $\|\mathbf{g}(\mathbf{y})\|_{\mathcal{D}} = 0$. In terms of equation (2.21) we have functions $f_0(x)$ that satisfy

$$\int_0^\infty k(x,y)\ f_0(x)\ dx = 0 \ . \tag{2.22}$$

These null-space functions ($f_0(x)$) can take *any* physical form that satisfies equation (2.22) and can be added arbitrarily to $\hat{\mathbf{f}}$ without affecting the data. Generally however they contribute a degree of ambiguity over the exact physical nature of the solution. So we have, from

the linearity of the integral

$$\int_0^\infty k(x,y)\ (f(x)+f_0(x))\ dx\ =\ g(y)\ +\ 0\ =\ g(y)\ . \tag{2.23}$$

The properties of the kernel function, $k(x,y)$, as mentioned earlier determine the *conditioning* of the problem once it is discretised to a matrix form like equation (2.10). A poorly conditioned kernel matrix has strong linear dependence in its rows and, in the worst case scenario, its determinant will tend towards 0 as the number of dependent rows increases, hence K will be singular (i.e. without an inverse). Equation (2.10) of Section 2.1.1.1 tells us that, provided the matrix K^{-1} exists, we formally have the exact solution $\mathbf{f} = K^{-1}\mathbf{g}$ to the Fredholm equation (of the first kind). However in many physical applications the matrix is very nearly singular and the inverse problem is considered to be *poorly conditioned*. The conditioning of a kernel operator (or matrix) is, as mentioned earlier, critical in monitoring error propagation from the observed data ($\mathbf{g}$) through to the recovered solution ($\mathbf{f}$).

In assessing the conditioning of the discretised inverse problem we anticipate a (vector) noise level ($\delta\mathbf{g}$) in our data measurements. We will observe error magnification in the solution ($\delta\mathbf{f}$) of the order

$$K\ \delta\mathbf{f}\ =\ \delta\mathbf{g}\ . \tag{2.24}$$

From this, implicitly using the 2-norm, we see that

$$\|\delta\mathbf{g}\|_2 \leq \|K\|_2\ \|\delta\mathbf{f}\|_2 \tag{2.25}$$

and we have reach a situation where we have, using the solution to equation (2.10), *if* the matrix K^{-1} exists (i.e. $\mathbf{f} = K^{-1}\ \mathbf{g}$)

$$\|\mathbf{f}\|_2 \leq \|K^{-1}\|_2\ \|\mathbf{g}\|_2\ . \tag{2.26}$$

Combining equations (2.25) and (2.26) we can see, on inspection, how errors in the data propagate through to the recovered solution

$$\frac{\|\delta\mathbf{f}\|_2}{\|\mathbf{f}\|_2}\ \leq\ \|K\|_2\ \|K^{-1}\|_2\ \frac{\|\delta\mathbf{g}\|_2}{\|\mathbf{g}\|_2}\ . \tag{2.27}$$

Thus, we define the *condition number* of the kernel matrix, C_K, $(1 < C_K < \infty)$ which can be expressed as $C_K = \|K\|_2\ \|K^{-1}\|_2 = \frac{\sigma_{max}}{\sigma_{min}}$. The second equality arises since the singular values of K^{-1} are just the reciprocal singular values of K, and shows that the spread of the singular values of K can disclose many hidden numerical problems, again see Chapter 5. Since the condition number controls the stability of the solution, consider the case when $\|\delta\mathbf{g}\| = 2\%$

then even a relatively well conditioned kernel with $C_K = 50$ will give rise to errors $\|\delta\mathbf{f}\|$ up to 100%. What is clear from the relationship above is the damaging effect of zero (or near-zero; as discussed above) singular values, or *eigenvalues*, because these will dramatically increase C_K and have, in general, highly oscillatory (often unphysical) eigenfunction counterparts (see the example on pg. 9 of Craig & Brown 1986 and discussions in Sections 2.1.3.2 and 2.1.3.1).

It is not only random errors in the data $\mathbf{g}$ that can cause serious numerical instability but also truncation and discretisation errors. Also, in certain cases K is *not* exactly known (i.e. it has its own associated error measure), so provided that we take $\mathbf{g}+\delta\mathbf{g} = (K+\delta K)(\mathbf{f}+\delta\mathbf{f})$ it is clear that errors in both K (δK) and $\mathbf{g}$ ($\delta\mathbf{g}$) we have

$$\frac{\|\delta\mathbf{f}\|_2}{\|\mathbf{f}\|_2} \leq \left(\frac{C_K}{1-C'_K}\right)\frac{\|\delta\mathbf{g}\|_2}{\|\mathbf{g}\|_2} + \left(\frac{C'_K}{1-C'_K}\right) \tag{2.28}$$

where we have defined $C'_K = \frac{\sigma'_{max}}{\sigma_{min}}$ and σ'_{max} is the maximum singular value of the δK matrix. Clearly this inequality requires that $0 \leq C'_K < 1$. Notice that in the case of $\delta K = 0$ ($C'_K \equiv 0$) this inequality reduces to that of equation (2.27). The study of inverse problems with kernel errors has been treated in only a limited number of cases (e.g., Goncharskii et al. 1972; Petrov & Khovanskii 1973) and even then only for very *small* errors δK.

This section has discussed some more of the components necessary to understand the ill-posed inverse problems presented in this thesis. The next section covers the application of such understanding to obtain unique numerically stable solutions.

2.1.3 Numerical solution of inverse problems: regularisation

The following sections discuss the use of *a priori* information to obtain a 'reliable' solution where reliable, in this sense, means numerically stable. We describe and demonstrate three of the most popular numerical techniques for solving inverse problems: Tichonov Quadratic Regularisation (QR), truncated Singular Value Decomposition (SVD) and the Maximum Entropy (ME) techniques. In Section 2.1.4 we will use all three formalisms to demonstrate their effectiveness in solving an inverse problem (one amenable to analytical solution) discussed in Rust & Burrus (1972).

We have already seen (from equations (2.10) and (2.11)) that the solution of an inverse problem (from a Fredholm integral equation) can be reduced to solving a 'simple' matrix equation. However, in Section 2.1.2 we discovered how the ill-posedness of inverse problems and the poor conditioning of the kernel matrix can wreak havoc if not considered carefully. Indeed, the fact that we 'know', or have theorised, how the solution will 'look' (e.g. many

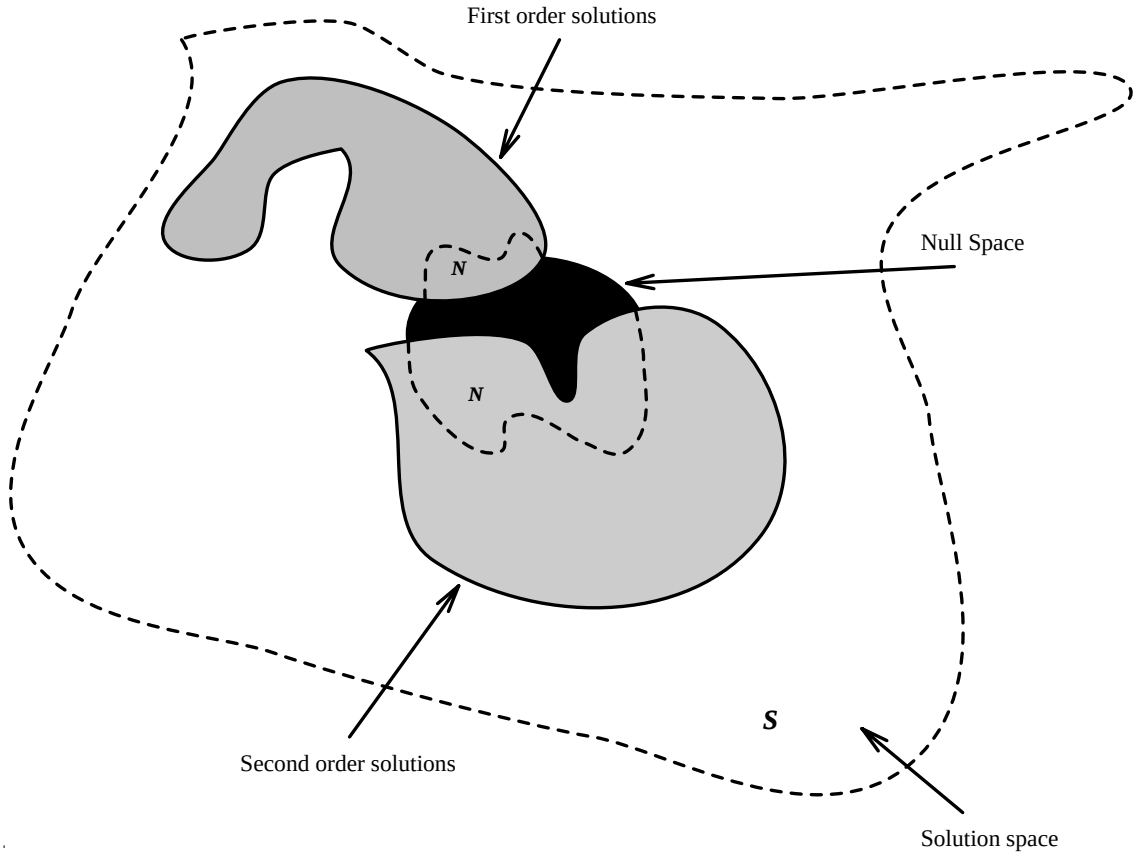

Figure 2.2: The hypothetical representation of the solution space $\mathcal{S}$ and the domains in which various types of characteristic solution belong. In particular we identify the null-space $\mathcal{N}$ and the domains of solutions with continuous first and second order derivatives that satisfy some criterion (see, e.g., equation(2.29)). The use of *a priori* information about the functional nature of the solution effectively constrains the solution to lie in one of these domains of $\mathcal{S}$.

physical situations require a positive and/or monotonic solution) means we can apply some *a priori* 'knowledge' (e.g. smoothness) to extract a particular 'unique' solution from the possibly infinite number of solutions in the unconstrained case (see, e.g., figure 2.2). As an example consider again the DEM function of Section 2.1.1.2 where *positivity* would seem like an appropriate constraint to apply to the solution (we do not want regions of the solar atmosphere having a imaginary electron density, $n_e^2 < 0$). In the language of the previous section, this only makes available the area of the solution space where $\|\mathbf{f}(\mathbf{x})\| > 0$ for all $\mathbf{x}$.

For completeness, and accuracy, the discretised integral equation $K\mathbf{f} = \mathbf{g}$ *should* be written as $K\hat{\mathbf{f}} = \hat{\mathbf{g}}$ where $\hat{\mathbf{f}}$ $(\mathbf{f}+\delta\mathbf{f})$ and $\hat{\mathbf{g}}$ $(\mathbf{g}+\delta\mathbf{g})$ are actual realisations of f and g. Classically, it is considered that a 'solution' (belonging to the solution space $\mathcal{S}$) satisfies $\|K\hat{\mathbf{f}} - \mathbf{g}\|_{\mathcal{D}} \leq \|\delta\mathbf{g}\|_{\mathcal{D}}$, for metric $\|.\|_{\mathcal{D}}$ in the data space $\mathcal{D}$. However, since $\|\mathbf{f} - \hat{\mathbf{f}}\|_{\mathcal{S}}$ may be arbitrarily large we must introduce a *regularisation* (or *trade-off*) constraint to eliminate the *very* large

number of high frequency solutions known to originate from small data perturbations without reducing the 'freedom' of the solution too much. This is a direct consequence of the *Riemann-Lebesgue* lemma - see, e.g., Sneddon (1972)

$$\int_a^b k(x,y)\; a_m \left\{ \begin{matrix} cos(my) \\ sin(my) \end{matrix} \right\} dy \to 0 \;\; \text{as } m \to \infty$$

which holds for *any* bounded square integrable ($\int\int |k(x,y)|^2 dxdy \leq M$) kernel. This shows that *any* Fourier amplitude a_m present in $f(y)$ is smoothed out by the action of the kernel, i.e. a_m may be smoothed out in the data, but its high frequency Fourier component is still present in the solution.

As may be guessed from the previous paragraph much of the methodology for solving inverse problems revolves around 'standard' least-squares minimisation procedures, i.e. we seek to reconstruct the observables ($\hat{\mathbf{g}}$) by 'suggesting' forms of the unknown ($\hat{\mathbf{f}}$) using as much *a priori* information as possible about the solution. So the method of Lagrange gives (minimising with respect to $\hat{\mathbf{f}}$)

$$\min_{\hat{f}} \; \|\mathbf{g} - K\hat{\mathbf{f}}\|^2 + \lambda \|\Phi(\hat{\mathbf{f}})\|^2 \tag{2.29}$$

where the functional $\Phi(\hat{\mathbf{f}})$ depends on the features we wish $\hat{\mathbf{f}}$ to exhibit a priori. Equation (2.29) introduces the constant λ, the Lagrange multiplier or *trade-off* parameter, which adjudicates a delicate compromise between 'good' recovery and domination by *a priori* information (e.g. see figure 2.3, Jin & Hou 1997 and others). The following subsections show how the regularised inversion process is carried out, at least in principle[4].

2.1.3.1 Quadratic regularisation

In most cases of practical interest, the solution $\mathbf{f} = K^{-1}\,\mathbf{g}$ of equation (2.10) is numerically unstable if the solution is not regularised. In an attempt to justify this statement we must construct a maximum likelihood, or least squares, solution of equation (2.10), i.e. we consider solving

$$\min_{\hat{\mathbf{f}}} \sum_{i=1}^{M} \left[g_i \; - \sum_{j=1}^{N} K_{ij}\, \hat{f}_j \right]^2 \tag{2.30}$$

where $\hat{\mathbf{f}}$ is now our estimate of the actual solution, $\mathbf{f}$. Differentiating this with respect to the kth component, $\hat{f}_k$, we obtain

$$\sum_{i=1}^{M} K_{ik} \left(g_i \; - \sum_{j=1}^{N} K_{ij} \hat{f}_j \right) = 0 \tag{2.31}$$

[4]Chapter 8 of Craig & Brown (1986) provides a useful recipe for how one should approach the inversion of remotely sensed data.

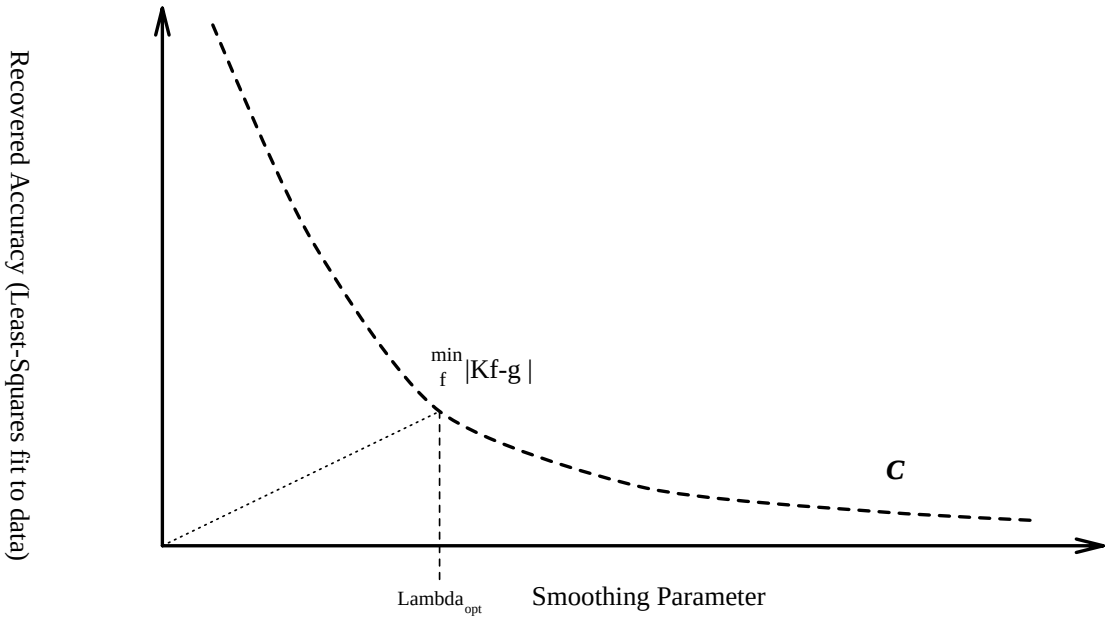

Figure 2.3: A typical plot of recovered accuracy ($|\mathbf{g} - K\ \hat{\mathbf{f}}|_{\mathcal{S}}$) against the smoothing parameter, λ. Clearly, the increase of the smoothing parameter does not improve the least-squares fit to the data. The point where the two balance is the point on the curve ($\mathcal{C}$) nearest to the origin ($\min_{\mathcal{C}} |\mathcal{C} - \mathbf{0}|$) is extrapolated to yield the ideal choice of λ.

which can also be re-written in matrix formulation as

$$\hat{\mathbf{f}} = \left(K^T\ K\right)^{-1} K^T \mathbf{g} \tag{2.32}$$

provided that $\left(K^T\ K\right)^{-1}$ exists. The fact that we must smooth $\hat{\mathbf{f}}$ arises because, in the process of inverting the matrix $K^T\ K$, small eigenvalues (singular values) can cause small variations in the data set to be magnified dramatically in the recovered solution, leading to the highly unstable and unphysical solutions.

Regularisation requires that the *a priori* information used to complete the definition of the inverse problem is a smoothness condition on the source function. So from the statement of equation (2.29) above, we obtain a *smooth* solution by bounding an appropriate *linear* functional, say $\mathcal{H}\hat{\mathbf{f}}$, subject to the classical constraint that $\|K\hat{\mathbf{f}} - \mathbf{g}\|$ is minimised. So we are reduced to solving, again using the $2-$ norm implicitly

$$\|K\hat{\mathbf{f}} - \mathbf{g}\|^2 + \|\mathcal{H}\hat{\mathbf{f}}\| = \min \tag{2.33}$$

where λ is the regularisation (or smoothing) parameter (cf. the Lagrange multiplier of above).

The inverse problem literature details the various forms of $\mathcal{H}\hat{\mathbf{f}}$: Tichonov (1963), whose name is synonymous with the method of regularisation, suggested a *zeroth* order approach

where $\mathcal{H} = \mathcal{I}$, the identity operator, whereas Phillips (1962) used a *second* order regularisation method. In other words he sought solutions that can 'fit the data' ($\|K\hat{\mathbf{f}} - \mathbf{g}\|^2 \leq \|\delta\mathbf{g}\|^2$) but were 'sufficiently smooth' (to eliminate highly oscillatory components) which minimise

$$\|\mathcal{H}\hat{\mathbf{f}}\|_2^2 = \|\hat{\mathbf{f}}''\|_2^2 = \int_a^b [\hat{f}''(y)]^2 \, dy \tag{2.34}$$

where $\hat{f}''(x)$ indicates double differentiation with respect to x. We will therefore restrict our study to first and second order discretised Quadratic Regularisation (QR) functions ($\Phi(\hat{\mathbf{f}}) = \mathcal{H}\hat{\mathbf{f}}$) respectively, which take the form

$$\Phi_1(\hat{\mathbf{f}}) = \sum_{j=1}^{M} \left(\hat{f}_{j+1} - \hat{f}_j\right)^2 \quad \text{and} \tag{2.35}$$

$$\Phi_2(\hat{\mathbf{f}}) = \sum_{j=1}^{M} \left(\hat{f}_{j+1} - 2\hat{f}_j + \hat{f}_{j-1}\right)^2 . \tag{2.36}$$

Equations (2.35) and (2.36) are forward difference estimates (see, e.g., Chapter 12 of Beyer 1991) of the first and second derivatives of $\hat{\mathbf{f}}$ and their respective operators $\mathcal{H}$ determine the region of the solution space in which $\hat{\mathbf{f}}$ can lie (see, e.g., figure 2.2).

Since our treatment focuses on the similarity between solution of inverse problems and the corresponding singular linear systems of equations, we cast equation (2.33) as a matrix system for $\mathcal{H}_2$, the second order regularising functional of equation (2.36). So, performing an analysis similar to the one prior to equation (2.32), differentiating equation (2.33) with respect to $\hat{f}_k$, we obtain the regularised matrix solution

$$K^T\hat{\mathbf{g}} = \left(K^T K + \lambda H\right)\hat{\mathbf{f}} , \tag{2.37}$$

where H is the *smoothing matrix*

$$H = \begin{pmatrix}
1 & -2 & 1 & & & & & & & & \\
-2 & 5 & -4 & 1 & & & & & & & \\
 & & & \ddots & & & & & & & \\
0 & \dots & 0 & 1 & -4 & 6 & -4 & 1 & 0 & \dots & 0 \\
 & & & & & & & \ddots & & & \\
 & & & & & & & 1 & -4 & 5 & -2 \\
 & & & & & & & & 1 & -2 & 1
\end{pmatrix}$$

and an *estimate* of λ is customarily taken to balance the bracketed term (i.e. $\lambda \sim \frac{tr(K^T K)}{tr(H)}$ where $tr(M)$ is the *trace* of the matrix M), or by obtaining a plot similar to figure 2.3.

We can also obtain the solution to equation (2.37) iteratively, using a method implemented by Tichonov (1963). So, if we substitute for K (in equation (2.10)) its *approximate inverse* (Tichonov 1963 and Louis 1996) $K' = K^T\ K\ +\ \lambda H$, then we are seeking to minimise the modulus of *step-wise difference vector* over $i = 1, \ldots, N$ (N is typically a large number)

$$\mathbf{r}_i = \mathbf{g} - K\mathbf{x}_i \tag{2.38}$$

where $\mathbf{x}_i$ is the solution at step i and is given by

$$\mathbf{x}_i = \mathbf{x}_{i-1} + K'\ K^T\ \mathbf{r}_{i-1}\ . \tag{2.39}$$

So, to help understand the regularisation process we look *symbolically* at equation (2.37) and use the Tichonov formalism. Therefore, by setting $H = I_n$ where I_n is the identity matrix of order n in equation (2.37) we have

$$\hat{\mathbf{f}}\ =\ \frac{\mathbf{K^T g}}{(\mathbf{K^T\ K}\ +\ \lambda \mathbf{I})}\ . \tag{2.40}$$

On expanding for $\hat{\mathbf{f}}$ in terms of the eigenvectors of $K^T\ K$ (the singular vectors $\mathbf{v_j}$ from above), we have

$$\hat{\mathbf{f}}\ =\ \sum_{\mathbf{j=0}}^{\infty}\ \left(\frac{\epsilon_\mathbf{j}}{\epsilon_\mathbf{j}^\mathbf{2} + \lambda}\ \mathbf{g_j}\right)\ \mathbf{v_j}\ . \tag{2.41}$$

From this relationship it is clear that small eigenvalues ($\epsilon_j^2 < \lambda$) are 'replaced' by λ in the calculation and their oscillatory counterparts are seen to be 'filtered out' since the bracketed term is $\ll 1$.

In short we can view the regularisation process for solution space, $\mathcal{S}$, and the regularised solution $\hat{\mathbf{f}}$ as actively

1. restricting $\hat{\mathbf{f}}$ to lie in a region of $\mathcal{S}$ only available to smooth functions of a particular nature, traditionally a polynomial of low order n or to a low order singular function expansion (see below).

2. Minimising the dimension of the null-space of the problem by removing the dependence on near-zero eigenvalues which in turn will,

3. filter out the high frequency components in $\hat{\mathbf{f}}$.

2.1.3.2 Singular Value Decomposition

When we have to consider systems of equations with singular, or numerically close to singular, matrices we have a very powerful ally in Singular Value Decomposition (SVD). This form

of decomposition can *always* be performed irrespective of how singular the matrix is, and is almost unique (Craig & Brown 1986). The SVD technique is used extensively, and has been essentially been optimised in the field of Helioseismology (see, e.g., Christensen-Dalsgaard et al. 1993, Schou et al. 1994, Hansen 1994 and Basu et al. 1997).

From Section 2.1.2, we have seen that any matrix, M, can be decomposed into a multiplication of two orthogonal matrices (U and V) and the diagonal matrix (Σ) which contains the singular values of M. Again, the SVD of M is

$$M \;=\; U\,\Sigma\,V^T\,, \tag{2.42}$$

and that of M^{-1} is given by

$$M^{-1} \;=\; V\,\Sigma^{-1}\,U^T \;=\; V\,[\mathrm{diag}(\frac{1}{\sigma_1},\ldots,\frac{1}{\sigma_n})]\,U^T \tag{2.43}$$

where we remember that $U\,U^T = V^T\,V = I$ and for a diagonal matrix the elements of its inverse assume their reciprocal values.

As justification of the above statement on uniqueness consider the discretised inverse problem of equation (2.10) which will possess a null-space (as described above). The non-uniqueness of the solution will depend critically on the number of zero (or zero to within numerical accuracy) singular values, with each one adding an extra dimension to the null-space (i.e. each contributing another linearly independent vector $\mathbf{f_o}$ satisfying $K\,\mathbf{f_o} = \mathbf{0}$), the dimension of the null-space is termed the *nullity*[5] of K. Although this sounds complicated, consider that the solution space has dimension N (the dimension of $\mathbf{f}$) and the *rank* of K (the dimension of the sub-space of $\mathbf{g}$ which is reached by the mapping of $\mathbf{f}$ by K) are simply related by the *rank and nullity theorem* which states that "rank plus nullity equals N".

It is simple to develop an expression for the solution of equation (2.10), $\hat{\mathbf{f}}$, in terms of the singular values, σ_i, and singular functions $\mathbf{u}_i$, $\mathbf{v}_i$ of the kernel matrix K from the relationships above. In any SVD reconstruction we have to *truncate* the singular values at some minimum 'cut-off' value which roughly corresponds to the choice of λ above. Typically this cut-off is chosen to reduce the effect of small (numerically very small or just small with respect to the noise level) or zero singular values. So we choose the cut-off for p where σ_{p+1} is less than the larger of the precision of the computer performing the calculation, or the data noise level ($\delta\mathbf{g}$). We then obtain the expression for $\mathbf{f}_p$, given by

$$\mathbf{f}_p \;=\; \sum_{j=1}^{p} \frac{g_j}{\sigma_j}\,\mathbf{v}_j \tag{2.44}$$

[5]indeed, if K is non-singular, we can assume that since the nullity is *zero* our solution is unique

or by direct analogy to equation (2.41)

$$\hat{\mathbf{f}} \;=\; \sum_{j=0}^{N} \left(\frac{\sigma_j}{\sigma_j^2 + \lambda}\, g_j \right) \mathbf{v}_j \;. \tag{2.45}$$

where we have replaced the need to truncate with a trade-off parameter, λ.

Although the solution returned may be a composite of infinitely many of these null-space functions, we must "choose" carefully the value of λ, or where we want to truncate our solution, using any *a priori* constraints we have decided to impose on the problem at hand. These *a priori* constraints only allow us to *select* an 'unique' solution.

2.1.3.3 Maximum Entropy

Maximum Entropy is one of the most commonly used techniques for stabilizing the solution of inverse problems. The approach is similar to that of Quadratic Regularisation, i.e. we seek the solution $\hat{\mathbf{f}}$ that minimises

$$\min_{\hat{\mathbf{f}}} \sum_{i=1}^{M} \left[g_i \;-\; \sum_{j=1}^{N} K_{ij}\, \hat{f}_j \right]^2 + \lambda \Phi(\hat{\mathbf{f}}) \tag{2.46}$$

where the smoothing functional now takes the non-linear form

$$\Phi(\hat{\mathbf{f}}) = \sum_{i=1}^{N} \hat{f}_i - m_i - \hat{f}_i \log(\frac{\hat{f}_i}{m_i}) \tag{2.47}$$

and m_i is some 'prior estimate' of $\hat{\mathbf{f}}$ towards which the function will be smoothed, commonly assumed to be flat (Twomey 1963). The *a priori* information used in a typical Maximum Entropy recovery is that each element of the solution is independent of any other element and so the 'smoothing' is applied to the solution in a global manner.

Although there are many analogies to Quadratic Regularisation when considering a Maximum Entropy technique, one advantage of using Maximum Entropy is that it allows an additional *a priori* constraint to be implied automatically, viz. ME will impose positivity. We have seen previously that positivity is very useful in many physical situations. Further analysis of the Maximum Entropy technique is beyond the scope of this discussion, because of the non-linearity of the smoothing operator. The ME algorithm used in the calculations of the example below and future chapters (unless otherwise stated) simply implements the GUIPS[6] package of routines.

[6] GUIPS is the acronym given to the Glasgow University Inverse Problem Software, a collection of routines to find solutions to ill-posed inverse problems. The routines were written by Dr. A. M. Thompson

2.1.4 A fully worked example

As an aid in the understanding of the discussion above, we will discuss the solution of a specific inverse problem. As an ideal test we will consider the solution of the Fredholm equation presented in Rust & Burrus (1972) (first discussed by Phillips 1962), the reason being that this is amenable to an *analytical* solution.

The inverse problem, stated specifically is

$$y(t) = \int_{-6}^{6} k(s,t)\; x(s)\; ds \;, \tag{2.48}$$

where the kernel function $k(s,t)$ ($|t| \leq 6$) is given by

$$k(s,t) = \begin{cases} 1 + \cos\left(\frac{\pi(s-t)}{3}\right) & \text{for } |\mathrm{s}-\mathrm{t}| \leq 6 \\ 0 & \text{for } |\mathrm{s}-\mathrm{t}| > 3 \end{cases} \tag{2.49}$$

and the data function[7]

$$y(t) = \begin{cases} (6-|t|)\left[1 + \frac{1}{2}\cos\left(\frac{\pi t}{3}\right)\right] + \frac{9}{2\pi}\sin\left(\frac{\pi t}{3}\right) & \text{for } |\mathrm{t}| \leq 6 \\ 0 & \text{for } |\mathrm{t}| > 6 \end{cases} \;. \tag{2.50}$$

Thus, to complete equation (2.48) we require the exact solution function $x(s)$ which is

$$x(s) = \begin{cases} 1 + \cos\left(\frac{\pi s}{3}\right) & \text{for } |\mathrm{s}| \leq 6 \\ 0 & \text{for } |\mathrm{s}| > 3 \end{cases} \;. \tag{2.51}$$

In order to test the GUIPS (and SVD) inversion techniques discussed above we must discretise the integrand of equation (2.48) into a matrix form (cf. equation (2.10)). To do this we invoke Simspon's extended quadrature rule (i.e. multiplying the rows of the kernel matrix by integration weights). The form of the kernel matrix can be observed at the top of figure 2.4 with its corresponding singular value *spread* below. We observe from the ratio of maximum to minimum singular values $\frac{\sigma_{max}}{\sigma_{min}}$ that the kernel matrix of the problem has a condition number of 42886.66. We would hence expect a certain amount of oscillation in the solution with an adequate recovery of the form of the solution function given that the data input given to the inversion techniques is randomly perturbed (about the $y(t)$ value) by $\pm 15\%$. Indeed, this is observed in figure 2.5 where we have plotted the analytic solution (equation (2.51)) with those obtained by the various methods outlined above.

[7] The version of $y(t)$ printed in the original manuscript has the $\frac{9}{2\pi}$ factor multiplying the wrong term in the equation, this is the amended version.

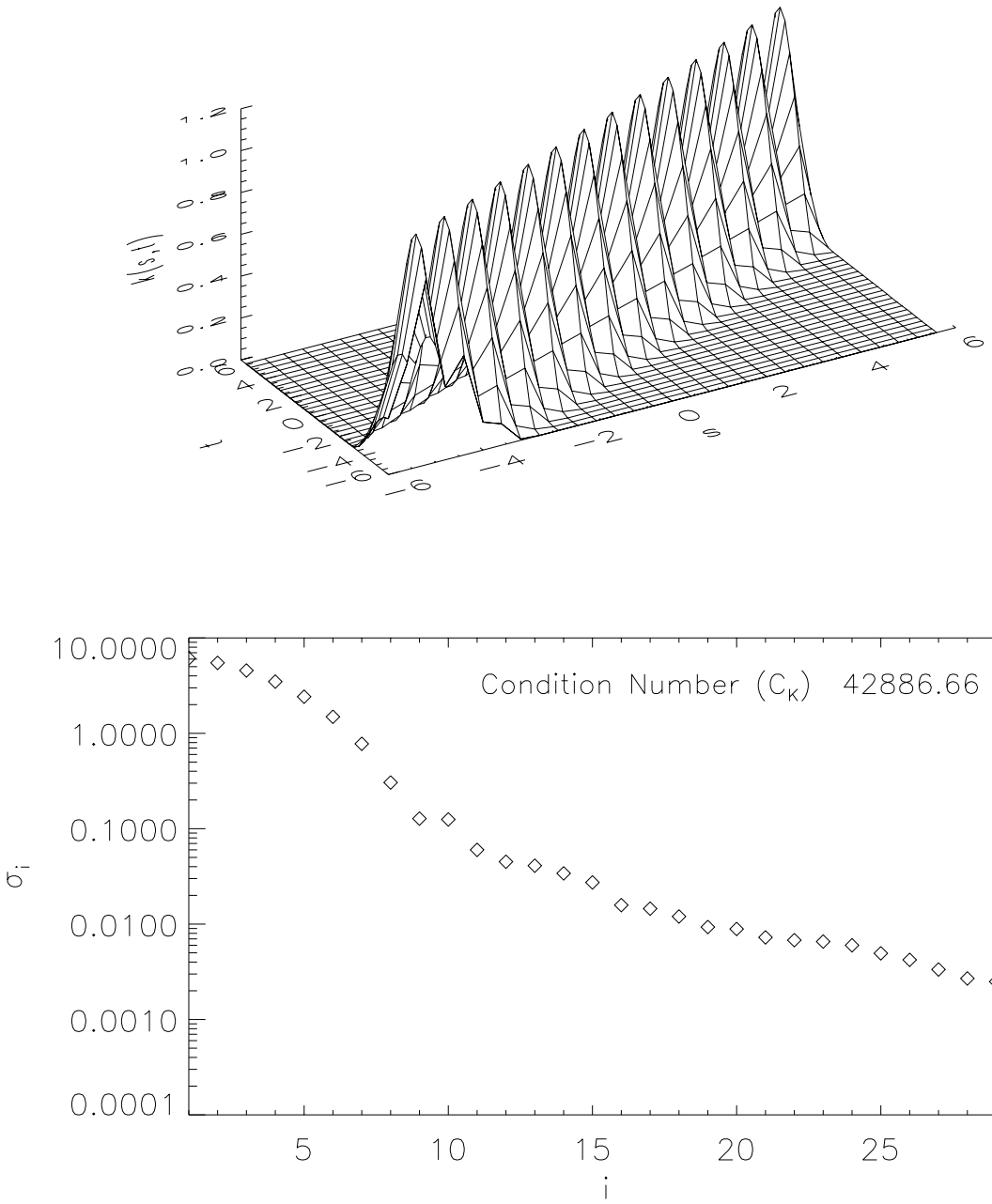

Figure 2.4: The surface representation of the kernel matrix (top) of equation (2.49) and the distribution in size of its singular values (bottom).

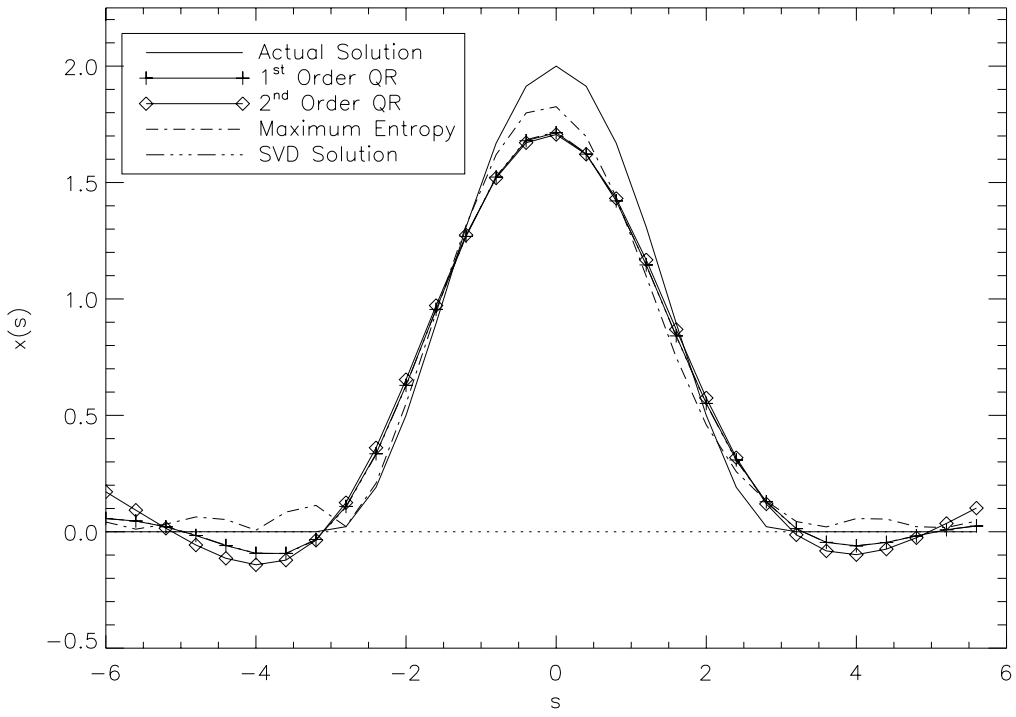

Figure 2.5: Results of the various inversion techniques (details in the figure legend) discussed in this chapter operating on the analytical inverse problem presented in Rust & Burrus (1972). Subtle differences are visible for each version of the solution and these difference are characteristics of those particular routines.

2.2 Atomic Physics

The principle aim of solar spectroscopy is to determine the characteristics of, and conditions within, the emitting plasma volume. However to understand the momentum and energy balance of the plasma we must determine the physical properties of the plasma from the data (e.g. the chemical abundance of the elements, the density, temperature, velocities and size of the emitting features). This requires adequate knowledge of the spectral formation process (i.e. we need a model to calculate the kernels discussed above). It is therefore necessary to combine the results of atomic physics with the study of analogous spectra in the laboratory (where we have 'full' control over the likely physical conditions) and then draw analogy to the physical conditions of the solar atmosphere through the knowledge acquired.

This section will explain the principle of spectral line formation in the distinctive regions of the solar atmosphere discussed previously in a heuristic[8] manner. Indeed such regions, particularly in the upper atmosphere (above the photosphere), depart significantly from local thermal equilibrium (LTE) and are known therefore as non-LTE plasmas. The 'coronal' regime in which we will work specifies that we will consider electrons as the only particles capable of collisionally exciting atoms to emit radiation. Such collisions are the dominant processes for populating the atomic levels for *allowed* (electric dipole), *intersystem* and *forbidden* transitions to be defined in due course.

Before we consider the formation of Ultraviolet (UV), extreme-UV (EUV) emission spectra and obtaining plasma diagnostics of the upper solar atmosphere we have to make our assumptions about the solar plasma clear to avoid ambiguity. The model of the upper solar atmosphere used throughout this thesis (cf. the temperature, density and pressure models presented of the previous chapter) is principally to make the calculation of atomic factors as simple as possible. Therefore we require that :

1. The plasma is *optically thin*.

2. Atomic hydrogen, the major constituent, must be fully ionised.

3. The electron distribution is Maxwellian in nature.

4. The abundances of the elements in the gas are constant.

[8]Heuristic in the sense that we will not worry about the physical processes behind the rate coefficients in the atomic rate equations for the time being, but will leave that for later discussion, see Chapter 6.

5. Including self-induced radiation (point 1), photo-excitation and de-excitation effects can be neglected.

So, taking all of these components together we can perform an analysis of the solar plasma similar to that of Pottasch (1964).

2.2.0.1 Features in atomic spectra

Spectroscopic studies of the light emitted or absorbed by atoms and ions from the early nineteenth century showed that each atom[9] emits a characteristic spectrum. Indeed, it was soon noticed that the spectrum itself was indicative of the electron structure of the atom. This eventually led to a better understanding of the periodic table through the X-Ray spectroscopy of Moseley (1913). The purpose of this short section is to introduce some spectroscopic terminology.

The spectrum emitted by neutral atoms of a given element, say X, is called the *first* spectrum of X and is denoted by X I; the spectrum emitted by the singly ionised X (i.e. X^{1+}) is called the *second* spectrum and is denoted by X II; and so on. It is then obvious that the number of line spectra that an atom is capable of producing is equal to its atomic number, Z. This means that we can observe spectra of H I, He I, He II, Li I, Li II, Li III, etc. Indeed, emission lines from Fe XXVI (*hydrogen-like iron*) have been observed in solar flare spectra, see Neupert et al. (1962), for example. Similarly, we would expect that ions with the same outer electron configurations have *similar* spectra; these are called *isoelectronic sequences* and are usually named according to the first neutral member (e.g., the lithium isoelectronic sequence is Li I, Be II, B III, C IV, N V, O VI, etc). Knowledge of such sequences allow us to associate plasma diagnostics to particular transitions of the *entire* isoelectronic sequence and is of particular use for probing the solar plasma because of the dependence of ionisation on temperature (i.e. $T_e \approx z^2 10^4$ K where z is the ionisation stage, $z = 1$ for a neutral atom) as we will see in later chapters.

2.2.1 UV/EUV spectral line formation

With the assumptions listed above, the total power P_l radiated in a particular spectral line labelled l, i.e. the atomic transition from level j to level i with respective energies E_j and E_i,

[9]The unqualified term "atom" will generally be used to mean either a neutral atom (a nucleus of charge $Ze+$ surrounded by N electrons, with N equal to Z) or a positively charged ion ($N < Z$).

from an optically thin plasma occupying a volume V is simply

$$P_l = \int\int\int_V h\nu_{ji} A_{ji} n_{u(l)} dV \quad \text{erg s}^{-1} \tag{2.52}$$

where h is Planck's constant, ν_{ji} is the frequency of the line, A_{ji} (s^{-1}) is the Einstein A-coefficient, and $n_{u(l)}$ (cm^{-3}) is the population density of the upper level $u(l) = j$. The expression for $n_{u(l)} = n_j$ can be decomposed, for simplicity, into

$$n_j = \frac{n_j}{n_{ion}} \cdot \frac{n_{ion}}{n_{el}} \cdot \frac{n_{el}}{n_H} \cdot \frac{n_H}{n_e} \cdot n_e \tag{2.53}$$

where $\frac{n_j}{n_{ion}} = f(n_e, T_e)$, $\frac{n_{ion}}{n_{el}} = g(T_e)$, $\frac{n_{el}}{n_H}$ and $\frac{n_H}{n_e}$ are the relative population of the upper atomic level of the line, the ionic abundance, elemental abundance, and relative abundance of H to electrons (having a value of 0.8 in the solar atmosphere) respectively. Full descriptions of these quantities can be found in Jordan (1969, 1970), Jacobs et. al (1977, 1980) and Arnaud & Rothenflug (1985). From this point on, or unless stated otherwise, we consider only the role of *bound-bound* (**b-b**) processes, i.e. those according to the $\frac{n_j}{n_{ion}}$ term.

The non-LTE rate equations for the coupling of levels j and i in a multi-level atom are

$$\frac{\partial n_i}{\partial t} + \mathbf{v} \cdot \frac{\partial n_i}{\partial \mathbf{x}} = \frac{D}{Dt} n_i = \sum_{j \neq i} n_j \mathcal{P}_{ji} - n_i \sum_{j \neq i} \mathcal{P}_{ij} \ , \tag{2.54}$$

where the term $\mathcal{P}_{ji} = R_{ji} + C_{ji}$ simply represents the total transition probability (s^{-1}) from level j to level i and is a sum of the radiative (R_{ji}) and collisional (C_{ji}) terms. $\mathcal{P}_{ji}$ is the probability of a large number of atoms in an ensemble making the transition from level j to level i. It is a function of time, peaking at $\mathcal{P}$, say, and is often interpreted as a transition 'rate' statistically applying to the whole ensemble and not to any individual atom. The radiative probabilities are $R_{ji} = A_{ji} + \tilde{J} B_{ji}$ where $\tilde{J}$ is the radiation field (implicitly taken to be **zero** in these calculations, item 5 above) and B_{ji} is the Einstein B coefficient of stimulated emission. The collisional rates are $C_{ji} = \sum_c n_c \langle v_c \sigma^{ji} \rangle$ where n_c is the number density of colliding particles which have a cross-section for making the transition from level j to i of σ^{ji}, and $n_c \langle v_c \sigma^{ji} \rangle$ is the probability integral involving the distribution function of the colliding particles. The transition rate per unit time is then

$$n_c \langle v_c \sigma^{ji} \rangle = n_c \int_0^\infty f(v_c) v_c \sigma(v_c)^{ji} \, dv_c \quad \text{s}^{-1} \tag{2.55}$$

where $f(v_c)$ is the velocity part of the distribution function. Since we are assuming that electron collisions will dominate, and electrons thermalise rapidly, the most likely distribution in this non-LTE low density regime will be the Maxwell-Boltzmann distribution. So the

collision probability per second for the de-excitation ($E_j > E_i$) takes the form

$$C_{ji} = \kappa \, \frac{\Upsilon_{ji}(T_e)}{g_j} T_e^{-\frac{1}{2}} \quad \mathrm{cm}^{-3}\ \mathrm{s}^{-1} \tag{2.56}$$

for T_e in degrees Kelvin where κ is a constant ($\kappa = 8.63 \times 10^{-6}$ for electrons) and g_j is the statistical weight of level j. The quantity $\Upsilon_{ji}(T_e)$ is known as the 'Maxwellian averaged collision strength' and is usually a smooth but weak function of temperature (see, e.g., Gabriel & Jordan 1971). The simple relationship between collisional excitation (C_{ij}) and de-excitation (C_{ji}) coefficients is then, using the *principle of detailed balance* ($n_i^* C_{ij} = n_j^* C_{ji}$ where n^* is the LTE population), given by

$$C_{ij} = C_{ji} \frac{g_j}{g_i} exp\left(\frac{-E_{ji}}{kT_e}\right) \quad \mathrm{cm}^{-3}\ \mathrm{s}^{-1} \tag{2.57}$$

where k is Boltzmann's constant and E_{ji} is the energy difference between levels j and i.

For a static medium ($\mathbf{v} \cdot \frac{\partial n_i}{\partial \mathbf{x}} = 0$) in statistical equilibrium ($\frac{\partial n_i}{\partial t} = 0$) the rate equations of equation (2.54) become

$$0 = \sum_{j \neq i} n_j \mathcal{P}_{ji} - n_i \sum_{j \neq i} \mathcal{P}_{ij} \tag{2.58}$$

and on substituting for $\mathcal{P}_{ji}$ and $\mathcal{P}_{ij}$ as above we have

$$0 = \sum_{j \neq i} n_j \left(A_{ji} + n_e C_{ji}\right) - n_i \sum_{j \neq i} \left(A_{ij} + n_e C_{ij}\right) \ . \tag{2.59}$$

However, this system of homogeneous equations is *not* closed, i.e. we require an equation to fix the set for n_i. Typically this is done by considering the abundance of the atom (Ab) such that

$$\sum_j n_j = Ab \cdot n_H \tag{2.60}$$

holds where n_H is the number density of Hydrogen. So, equations (2.59) and (2.60) form a closed linear system (e.g. $P\mathbf{n} = \mathbf{b}$) which can be solved for the atomic level populations $\mathbf{n}$, given P, the matrix of transition probabilities, and $\mathbf{b} = (0, \ldots, Ab \cdot n_H)$. Thus, we have prescribed the current state of the atom for the assumptions made earlier.

Now, we concentrate on particular transitions *within* an atom and we begin with the simplest case, a *resonance line*. A resonance line is one arising from allowed transitions from levels collisionally excited from the ground state to the ground state. We can consider the atom as a simple 3-level model (see figure 2.6). The solution of the statistical equilibrium equations is, for a transition from level j to level i,

$$n_e n_i C_{ij} = n_j \left(A_{ji} + n_e C_{ji}\right) \tag{2.61}$$

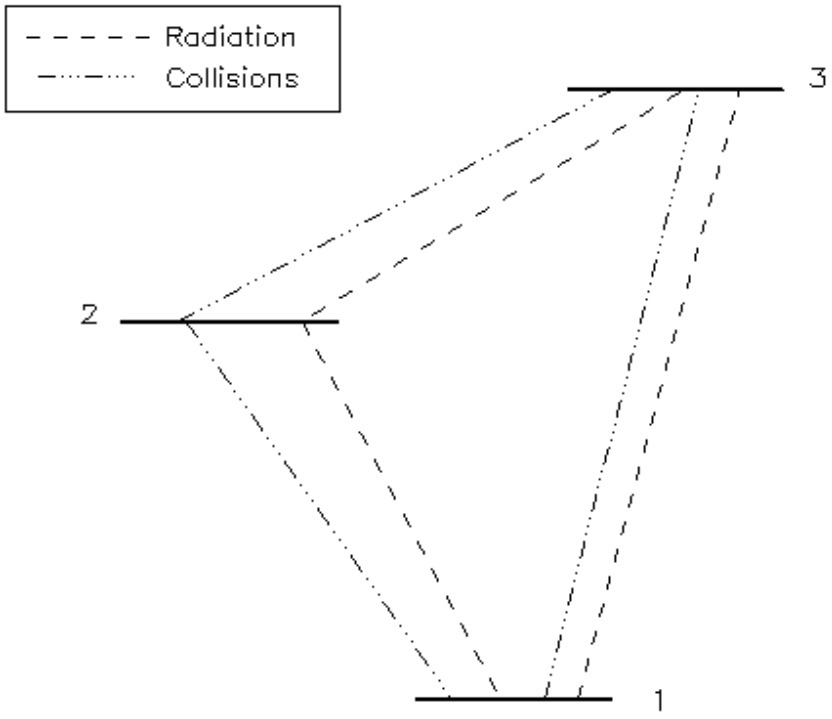

Figure 2.6: Schematic atomic ion containing just three bound levels. In this model, as indicated above, only spontaneous radiative decays and collisions of the atom with electrons are considered.

where we note that $A_{ji} > n_e C_{ji}$ and $n_i = n_{ion}$.

It is standard practice to define a quantity $K_l(n_e(\mathbf{r}), T_e(\mathbf{r}))$ called the line emission coefficient, or emissivity, normalised to the electron density squared of the transition. This is given by (cf. equations (2.52) and (2.53))

$$K_l(n_e, T_e) = \frac{h\nu_{ji} A_{ji}}{4\pi} \frac{n_j}{n_{ion} n_e} \frac{n_{ion}}{n_{el}} \frac{n_{el}}{n_H} \frac{n_H}{n_e} \quad \text{erg cm}^3 \text{ sr}^{-1} \text{ s}^{-1} \tag{2.62}$$

such that equation (2.52) becomes

$$P_l = 4\pi \int\int\int_V K_l(n_e, T_e) n_e^2 \, dV \quad \text{erg s}^{-1} \tag{2.63}$$

where l is simply a label replacing the combination ji. The importance of $K_l(n_e, T_e)$ will be seen below as its dependence on density and temperature will help yield diagnostics of the emitting plasma. However, this is an appropriate juncture to explain how emission lines are classified into three distinctive groups according to how their upper level is populated (see, e.g., Dere & Mason 1981 and Mason & Monsignori-Fossi 1994) :

1. Allowed lines that are collisionally excited just from the ground level - resonance lines - whose line emission coefficients are proportional to n_e^2 (cf. equation (2.63)), with $f(n_e, T_e)$ of equation (2.53) essentially independent of n_e.

2. Forbidden or intersystem lines with upper levels that are metastable. The radiative decay rates of these lines are so small that the electron collisions compete as a depopulating mechanism. The population of the metastable levels from which these lines originate and their intensity behaviour fall into three stages of development, depending on the electron density :

   - When the density is low, radiative decay dominates and the line intensity has a similar behaviour to that of an allowed resonance line (i.e. $\propto n_e^2$).
   - At intermediate densities ($n_e \sim \frac{A_{ji}}{C_{ji}}$; the *critical density*) the two mechanisms are competing and the population of the metastable level becomes important and the line intensity is proportional to n_e^δ, where $(1 < \delta < 2)$.
   - For higher electron densities the collisional process dominates and the metastable level attains Boltzmann equilibrium and the intensity varies as n_e.

   These ranges of electron densities are dependent on atomic parameters and differ for individual ions and transitions.

3. The intensities of allowed lines that are excited from low-lying metastable levels. Their intensities are dependent on the population of the metastable level from which they are excited. Once these levels attain a 'reasonable' population, but not its Boltzmann value, the line intensity will vary as n_e^{δ} $(2 < \delta < 3)$. When the Boltzmann level is reached the intensity varies as n_e^2.

2.2.1.1 Differential Emission Measures-DEMs

It is useful to define another important diagnostic tool at this point; the Differential Emission Measure (DEM) function, which we define by recalling equation (2.63) (incorporating dependence on $\mathbf{r}$)

$$P_l = 4\pi \int_V K_l(n_e(\mathbf{r}), T_e(\mathbf{r}))\, n_e^2(\mathbf{r})\, d^3\mathbf{r} \quad \text{erg s}^{-1}. \tag{2.64}$$

This equation, with full dependence on n_e and T_e included in the emission coefficient, was studied by formulating the integrand in terms of a function of electron density and temperature (Jefferies et al. 1972a, b). This function was later identified (see Brown et al. 1991, hereafter BDSA) as the bivariate DEM function of n_e and T_e, namely $\mu(n_e, T_e)$. Following the derivation of BDSA we make the following change of integration variable in equation (2.64):

$$d^3\mathbf{r} = \frac{dn_e dT_e}{|\nabla n_e|\,|\nabla T_e|\; sin\theta_{n_e,T_e}}\, dL_{n_e,T_e} \quad \text{cm}^3 \tag{2.65}$$

Hence, reducing the volume integral of equation (2.64) to a line integral of the emissivity along a line of constant n_e, T_e. Here θ_{n_e,T_e} (> 0) is the local angle between vectors ∇n_e and ∇T_e normal to surfaces S_{n_e}, S_{T_e} of constant electron density and temperature respectively[10], see figure 2.7. So, for every transition from level j to level i we have

$$P_l = 4\pi \int_{T_e}\int_{n_e} K_l(n_e, T_e)\mathrm{M}(n_e, T_e)\, dn_e dT_e \quad \text{erg s}^{-1} \tag{2.66}$$

where, from BDSA, $\mathrm{M}(n_e, T_e)$ is defined as

$$\mathrm{M}(n_e, T_e) = \oint_{L_{n_e,T_e}} \frac{n_e^2}{|\nabla n_e|\,|\nabla T_e|\; sin\theta_{n_e,T_e}}\, dL_{n_e,T_e} \quad \text{K}^{-1} \tag{2.67}$$

Usually, one does not directly observe the total radiated power P_l, but the intensity, $I_l = P_l/(4\pi S)$, where S is the area of the projected volume V. Defining $\mu(n_e, T_e) = \mathrm{M}(n_e, T_e)/S$, which has units of cm^{-2} K^{-1}, we find

$$I_l = \int_{T_e}\int_{n_e} K_l(n_e, T_e)\mu(n_e, T_e)\, dn_e dT_e \quad \text{erg cm}^{-2}\ \text{sr}^{-1}\ \text{s}^{-1}\ . \tag{2.68}$$

[10] Noting that this is the solar atmosphere. 'Standard' solar models which are very simplistic (see the example in Chapter 1 -figure 1.1). In a more realistic solar model we must accomodate regions where θ_{n_e,T_e} is zero and the temperature and density gradients are parallel.

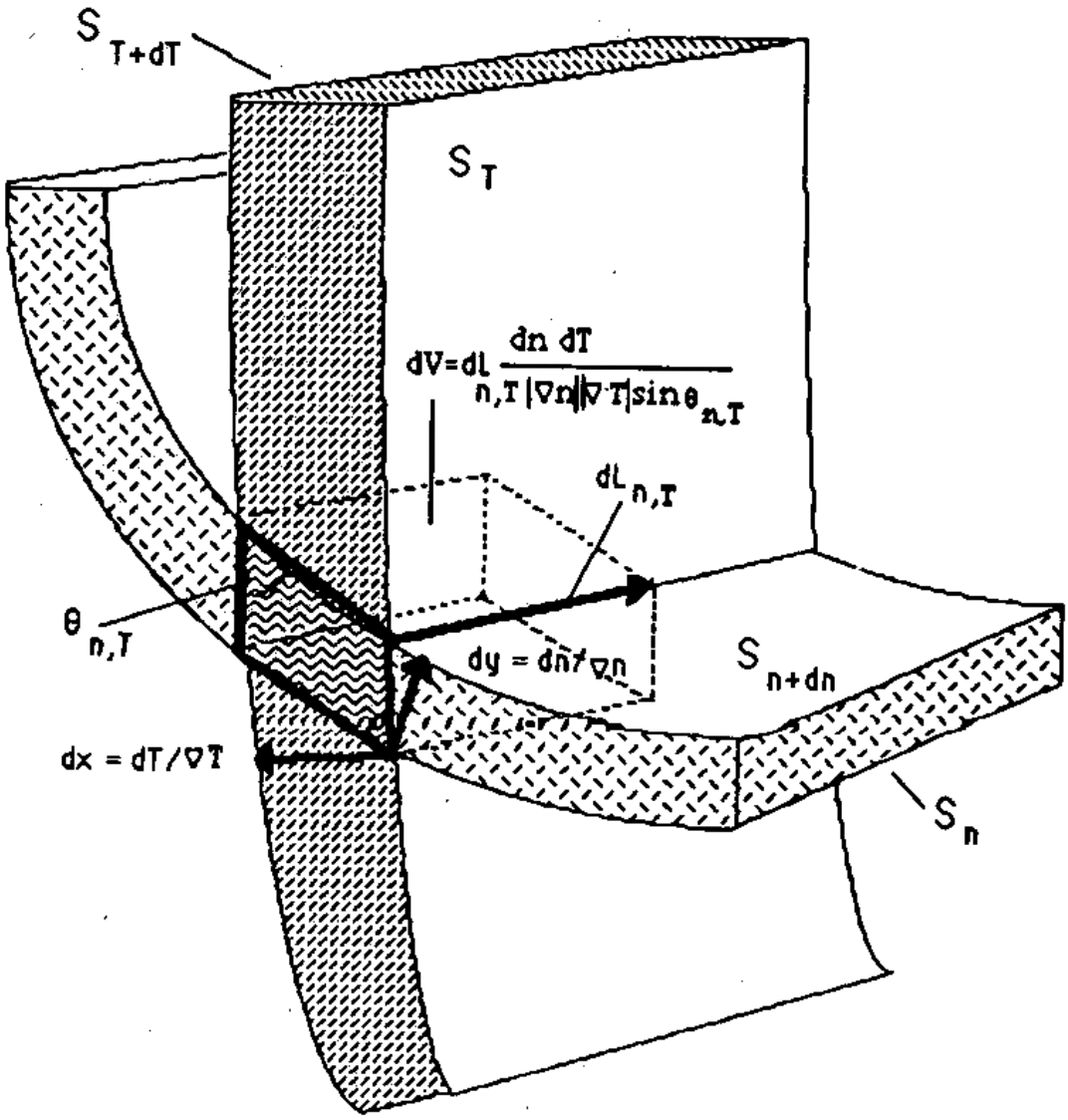

Figure 2.7: Two surfaces of constant temperature S_{T_e} and of constant density S_{n_e} intersecting on a line $L_{n_e,T_{n_e}}$. $\theta_{n_e,T_{n_e}}$ is the angle between the vectors ∇T_e and ∇n_e normal to the surfaces S_{T_e}, S_{n_e} respectively (taken from BDSA).

We are now in a position to define the differential emission measure in n_e, $\zeta(n_e)$, as the reciprocal density gradient weighted mean square electron density and, correspondingly the differential emission measure in T_e, $\xi(T_e)$, as the reciprocal temperature gradient weighted mean square electron density. These more intuitive functions (in terms of diagnostics at least) are obtained from equation (2.67) as follows:

$$\zeta(n_e) = \int_{T_e} \mu(n_e, T_e) dT_e \quad \text{cm}^{-2} \tag{2.69}$$

$$\xi(T_e) = \int_{n_e} \mu(n_e, T_e) dn_e \quad \text{cm}^{-5}\ \text{K}^{-1} \tag{2.70}$$

Thus, in terms of physical interpretation of a set of frequency integrated line intensities I_l alone, the differential emission measures in n_e and T_e form the spectroscopic basis for further interpretation of the raw data.

2.2.2 Plasma diagnostics

In the previous section we saw how to classify the majority of spectral lines observed in the upper solar atmosphere. In this section we present a double-edged description of obtaining useful diagnostics of the emitting plasma; obtaining electron temperatures and densities. The first definition is a purely heuristic, giving pictorial evidence to suggest that specific diagnostics occur in each of the iso-electronic sequences mentioned above whereas, the second is a much more mathematical description of obtaining a 'good' diagnostic and will be of considerable use in later chapters.

2.2.2.1 Electron temperature determination

It is important when attempting to infer the electron temperature (T_e) to remember that the plasma being observed is inhomogeneous and non-isothermal. Therefore, as is suggested by the integral above, the contribution to a particular line intensity comes form a wide range of densities and temperatures. The method described here will not directly allow the diagnosis of the true inhomogeneity of the plasma present, say in a solar flare where it is very possible that most, if not all, ionisation stages are emitting in a very small volume of plasma. However, this treatment will allow us to make an quantitative estimate of the 'mean' electron temperature of the plasma.

Several authors (Gabriel & Jordan 1969; Munro et al. 1971; Gabriel & Jordan 1971; Dere & Mason 1981; Doschek 1987; Mason & Monsignori-Fossi 1994) have developed and been actively using a technique involving two optically thin resonance lines with significantly

different excitation energies since the development of UV/EUV spectroscopy in the early 1960s. Consider initially an isothermal plasma of electron density n_e and volume V. The ratio of two resonance lines originating in levels 2 and 3 and decaying to the ground state, say level 1 (see figure 2.10 with level 3 not metastable and compare with figure 2.8), is given by

$$\frac{P_3}{P_2} = \frac{E_{13}}{E_{12}} \cdot \frac{C_{13}}{C_{12}} \tag{2.71}$$

where $E_{13} = h\nu_{13}$ and $E_{12} = h\nu_{12}$. So using equation (2.57) to substitute for the collisional excitation terms above to find that

$$\frac{P_3}{P_2} = \frac{g_2}{g_3} \frac{E_{13}\Upsilon_{13}(T_e)}{E_{12}\Upsilon_{12}(T_e)} exp\left(\frac{-(E_{13}-E_{12})}{kT_e}\right) . \tag{2.72}$$

It is clear from this equation, given the Υ's and E's, that this ratio is *only* dependent on the temperature and that this dependence comes almost entirely from the exponential term and in particular when $|\frac{E_{13}-E_{12}}{kT_e}| \gg 1$. As stated above this measure is *not* an adequate measure of T_e because we have explicitly assumed the plasma to be isothermal which is clearly *not* the case in the solar atmosphere, see Litwin & Rosner (1993) for physical discussion, or *any* image of the UV solar atmosphere. Another more practical drawback of this method is due to the fact that we require $|\frac{E_{13}-E_{12}}{kT_e}| \gg 1$ which implies a large wavelength separation of the lines can make observation and line calibration difficult.

It is possible to define mathematically this heuristic estimate of the electron temperature in terms of $\xi(T_e)$, by considering a line labelled i for which $K_i(n_e, T_e)$ is a weak function of density, such as a resonance line. $K_i(n_e, T_e)$ can then be replaced by $K_i(T_e) = K_i(n_e = n_0, T_e)$ and so we have, from equations (2.68) and (2.69)

$$I_i = \int_{T_e} K_i(T_e)\xi(T_e)dT_e . \tag{2.73}$$

For two such lines i and j, whose emission coefficients have different functional dependence on T_e (generated by the exponential term above), the ratio of the two line intensities is then given by

$$R_{ij} = \frac{I_i}{I_j} = \frac{\int_{T_e} K_i(T_e)\,\xi(T_e)\,dT_e}{\int_{T_e} K_j(T_e)\,\xi(T_e)\,dT_e} . \tag{2.74}$$

If the plasma were homogeneous we could express $\xi(T_e)$ function as $\xi(T_e) = \xi_0\,\delta(T_e - \langle T_e \rangle)$ such that, on substituting this expression into to equation (2.74) and integrating over the whole temperature domain, we have

$$R_{ij} = \frac{\xi_0\,K_i(\langle T_e \rangle)}{\xi_0\,K_j(\langle T_e \rangle)} \tag{2.75}$$

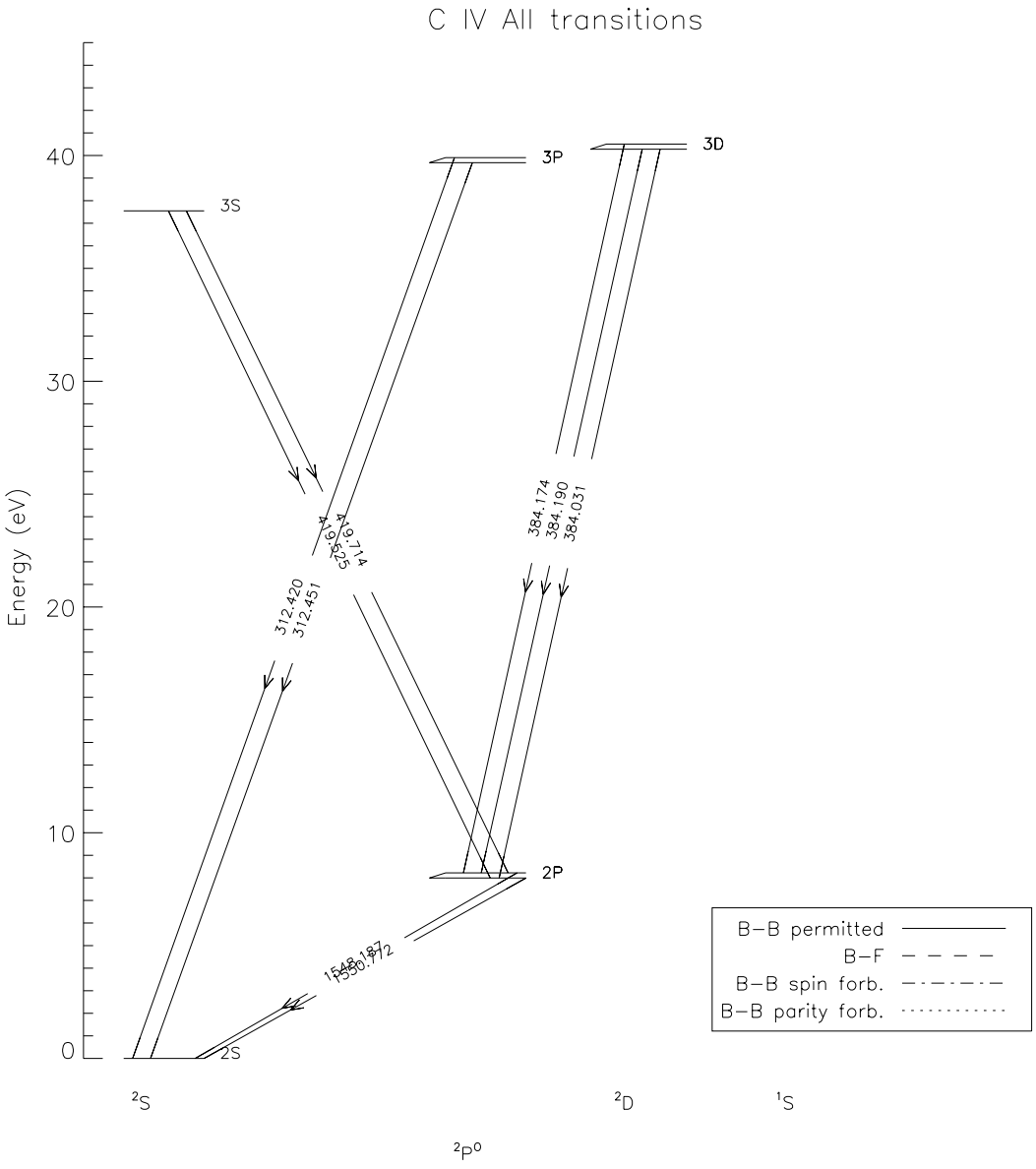

Figure 2.8: Grotrian diagram (Grotrian 1928) of all the atomic transitions for *lithium-like* C IV showing all the transitions (mean wavelengths for multiplets) in the wavelength range of the SOHO SUMER and CDS ($150 - 1610$Å) spectrometers.

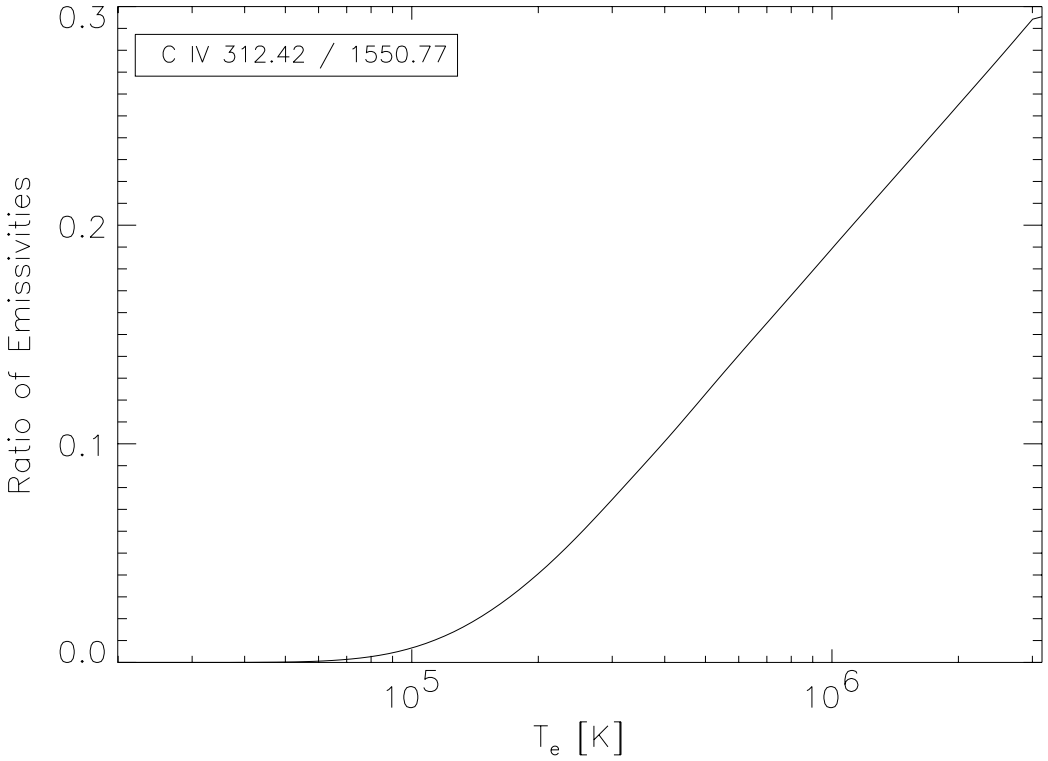

Figure 2.9: Plot of the temperature sensitive ratio of two resonance lines $\left(\frac{\lambda_{312.42}}{\lambda_{1550.77}}\right)$ of C IV. A value of 0.15 for the line intensity ratio will yield a mean spectroscopic temperature $\langle T_e \rangle$ of approximately 6×10^5 K.

and on dividing throughout by ξ_0 we may express R_{ij} in terms of the 'mean' spectroscopic temperature, $\langle T_e \rangle_{ij}$, for the particular line pair (i, j), *i.e.*

$$R_{ij} = \frac{K_i(\langle T_e \rangle)}{K_j(\langle T_e \rangle)} = S_{ij}(\langle T_e \rangle_{ij}) \tag{2.76}$$

where $S_{ij}(T_e) = \frac{K_i(T_e)}{K_j(T_e)}$ is a monotonic, bijective (invertible) function, that has a unique inverse on the temperature domain considered when we restrict our study to resonance lines only. Therefore, on inspection, the relation between $\langle T_e \rangle_{ij}$ and the observed line ratios R_{ij} is given by

$$\langle T_e \rangle_{ij} = S_{ij}^{-1}(R_{ij}) \tag{2.77}$$

and can be represented pictorially in figure 2.9.

2.2.2.2 Electron density determination

The ratio of emission lines with different density dependence has been widely used as a diagnostic of the electron density in the inhomogeneous solar atmosphere. However, Almleaky

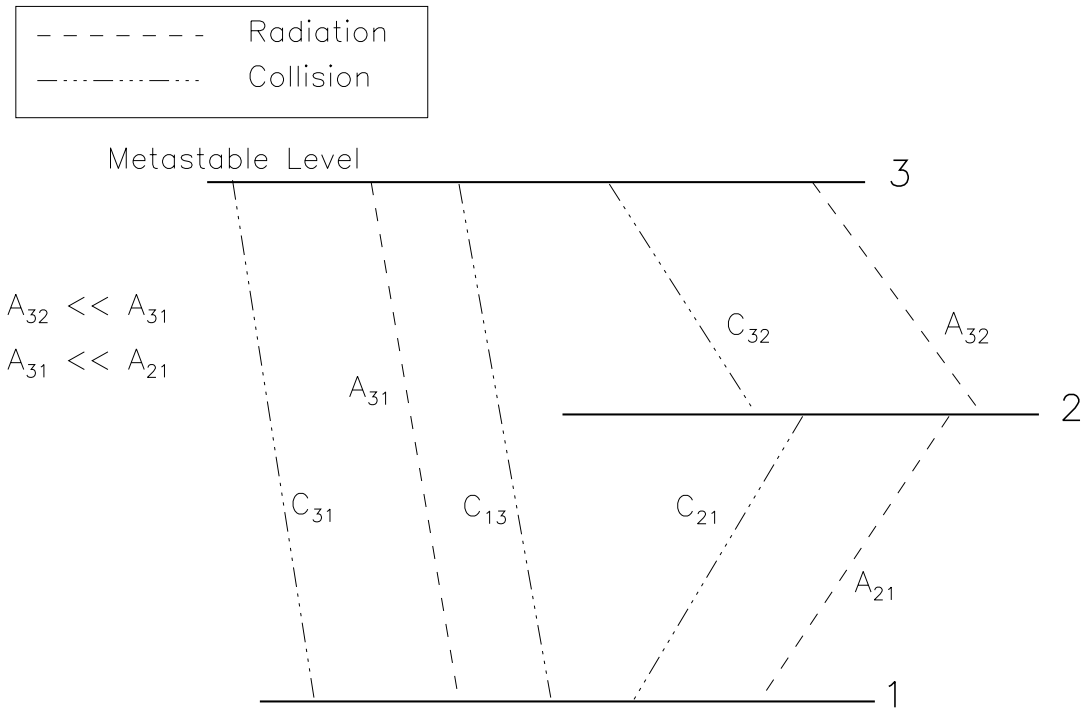

Figure 2.10: Schematic atomic ion containing just three bound levels (cf. figure 2.6). In this model we consider the additional effect when level 3 is metastable.

et al. (1989) and, more recently, Judge et al. (1997), have shown that such estimates are highly ambiguous. The following is a description of some of the basic principles used to obtain such an estimate. The treatment above has dealt solely with resonance lines and we have shown that the ratio of their line strengths is solely dependent on temperature. However, for an intersystem or forbidden line the upper level of the transition is a metastable (m) long--lived ($A_{mi} \approx C_{mi}$) level where the population is comparable to that of the ground level. Therefore atoms with metastable levels provide a good source of density diagnostics since the populations of the other levels in the atom are affected in a delicate balance by their presence.

So, considering figure 2.10 where we have a simple atom with level 3 a long-lived metastable level and looking at the ratio between transitions from levels 1 to 2 (a resonance line) and levels 1 and 3 (an intersystem line), we have

$$n_2 A_{21} = n_e n_1 C_{12} \quad \text{and} \tag{2.78}$$

$$n_3 \left(A_{31} + n_e C_{23}\right) = n_e n_1 C_{13} \; . \tag{2.79}$$

We then have for the ratio

$$\frac{P_2}{P_3} = \frac{C_{12}}{C_{13}} \cdot \frac{E_{12}}{E_{13}} \cdot \left(1 + \frac{n_e C_{23}}{A_{21}}\right) , \tag{2.80}$$

which has the same factor $(\frac{C_{12}}{C_{13}})$ as the temperature case, but the density dependence arises from the factor in brackets and especially when $n_e C_{23} \approx A_{21}$, the density at which this occurs is known as the "critical density".

Again, we can put this statement on a mathematical footing; we use an analogous approach to that of Almleaky et al. (1989). Consider an optically thin plasma that is isothermal with $T_e = T_0$. The total emission of a line labelled i, given by equation (2.68) and equation (2.70) is

$$I_i = \int_{n_e} K_i(n_e)\, \zeta(n_e)\, dn_e, \tag{2.81}$$

for $K_i(n_e) = K_i(n_e, T_e = T_0)$. Since the plasma has no unique n_e, we can nevertheless define a spectroscopic 'mean' electron density for the any ratio of lines displaying some degree of density sensitivity, for instance using a resonance line and an intersystem line from a common ionisation stage of a particular atom, as above. For this pair (i, j), we seek the electron density of a homogeneous plasma that would yield the same line ratio, R_{ij}, as the inhomogeneous plasma under observation. To achieve this we define $\zeta(n_e) = \zeta_0\, \delta(n_e - \langle n_e \rangle)$, where $\langle n_e \rangle$ is the 'mean' spectroscopic electron density as defined earlier. Given this, we follow the steps leading to equation (2.77) where we now have

$$\langle n_e \rangle_{ij} = G_{ij}^{-1}(R_{ij}) , \tag{2.82}$$

where G_{ij} is an invertible function (cf. S_{ij} of equation (2.77)) in the plasma regime we are considering. Again, this process can also be represent pictorially, see figure 2.11.

2.2.3 The nature of errors in line emissivities

We have seen that, for resonance lines, the line emissivity (equation (2.62)) for a line labelled i can be expressed as

$$K_i = \Upsilon_i\, \mathcal{X}_i\, \mathcal{Y}_i \tag{2.83}$$

where Υ_i is the Maxwellian-averaged collision strength, as above. We have simplified the components, made use of equation (2.61), introduced $\mathcal{X} = \frac{n_{ion}}{n_{el}}$ the ionisation fraction and $\mathcal{Y} = \frac{n_{el}}{n_H}$ the abundance of the element relative to hydrogen. To obtain an error estimate δK_i for K_i we note that

$$\frac{\delta K_i}{K_i} = \sqrt{\left(\frac{\delta \Upsilon_i}{\Upsilon_i}\right)^2 + \left(\frac{\delta \mathcal{X}_i}{\mathcal{X}_i}\right)^2 + \left(\frac{\delta \mathcal{Y}_i}{\mathcal{Y}_i}\right)^2} \quad , \tag{2.84}$$

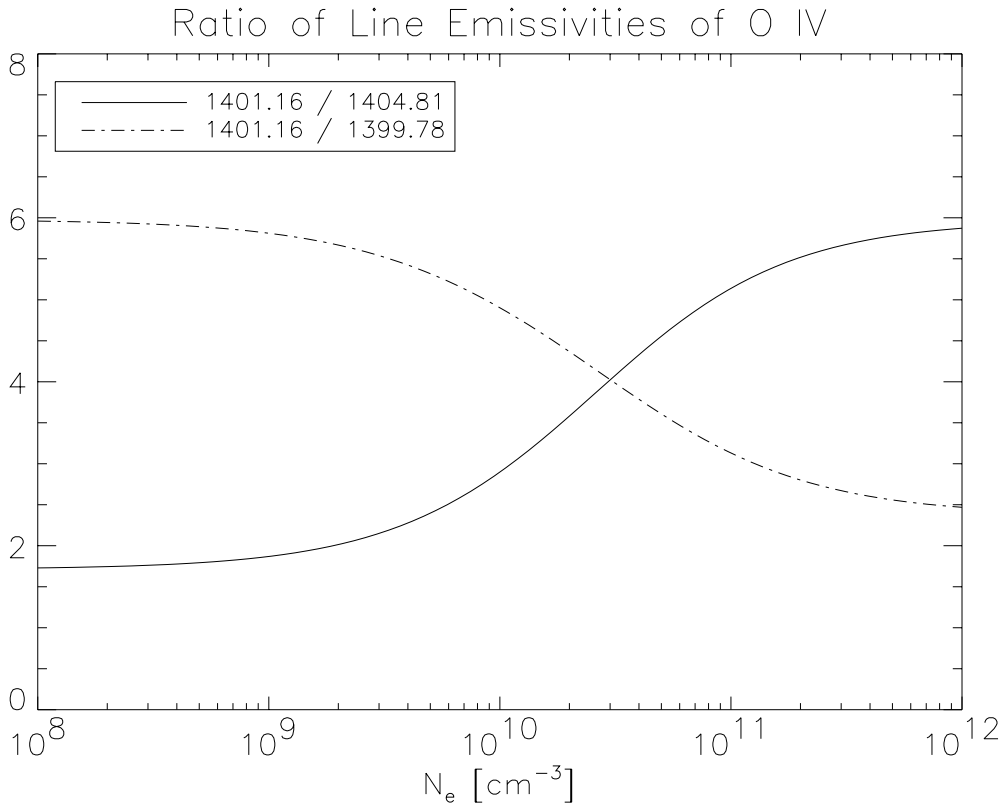

Figure 2.11: Plot of the density sensitive ratios $\left(\frac{\lambda_{1401.16}}{\lambda_{1404.81}}, \frac{\lambda_{1401.16}}{\lambda_{1399.78}}\right)$ of O IV. A value of 4.0 for both line intensity ratios will yield mean spectroscopic density $\langle n_e \rangle$ of approximately 3×10^{10} cm^{-3}.

and remembering that we have previously assumed that the elemental abundances are constant, the emissivity errors are dominated by a elementary factors, and they are (Judge et al. 1997) :

- Errors in electron-ion excitation cross-sections ($\delta\Upsilon_i$) depend on the isoelectronic sequence to which the ion belongs. They also depend critically on the type of transition (permitted, forbidden, etc.), relativistic effects and the assumptions made when calculating the collisional cross-sections (cf. the assumptions we made earlier). A recent laboratory study of the resonance lines of C IV (see figure 2.8) measured cross-sections with an accuracy of $\pm 7\%$ (Savin et al. 1995).

- Errors in the ionisation balance ($\delta\mathcal{X}_i$) that depend, not only the cross-sections but the structure of the emitting plasma (departures from equilibrium, Judge et al. 1995, of equation (2.54)) itself. These errors, systematic in nature for a particular ion, are likely to be of the order $\pm 20\%$. If non-equilibrium processes are present they can be *much* higher (P. G. Judge - Private Communication).

When considering transitions involving metastable levels calculation of the Einstein A--coefficients is important (cf. the statement after equation (2.80)) and from these arise an additional source of error. So, given equation (2.79) we have, making simplifications similar to those above

$$\frac{\delta K_i}{K_i} = \sqrt{\left(\frac{\delta\Upsilon_i}{\Upsilon_i}\right)^2 + \left(\frac{\delta\mathcal{X}_i}{\mathcal{X}_i}\right)^2 + \left(\frac{\delta\mathcal{Y}_i}{\mathcal{Y}_i}\right)^2} + \sqrt{\left(\frac{n_e}{n_{e_c}+n_e}\right)^2 \left(\left(\frac{\delta A_i}{A_i}\right)^2 - \left(\frac{\delta\Upsilon_i}{\Upsilon_i}\right)^2\right)} \quad (2.85)$$

where n_{e_c} is the aforementioned critical density; for $n_e \ll n_{e_c}$ this equation reduces to the form of equation (2.84).

Given this information Judge et al. (1997) conclude that errors in the line emissivities range upward from $\pm 30\%$ and are systematic in nature. The systematic nature of errors we will use to our advantage in the analysis of Chapter 4. However, from these statements it might be reasonable to ask " Why do we not simply set up a laboratory and measure the cross-sections needed to solve equation (2.54) directly ?". There are several reasons for this, and they are :

1. The number of cross-sections required for reliable determination of the emission coefficients scales as $kn(n-1)$ where k is a constant between one and two and n is the number of bound levels in the model.

2. Each cross-section has to be determined at all energies with the velocity distribution function at an energy resolution sufficient to allow accurate calculation of the rate coefficient.

3. In the solar atmosphere the atomic collision cross-sections are needed in the limit $kT_e \ll E_j - E_i$ and since the kinetic energy of the impacting particle is much less than the energy to make the transitions. Performing such experiments at low energies makes the system susceptible to external (electric and magnetic) effects.

4. The lifetimes of some of the atomic levels are too short to allow measurement (at all, in some cases) of the rate coefficients.

5. The production and containment of extremely highly ionized species (e.g., those more than three times ionised found in regions of the solar corona) in a laboratory plasma is difficult.

Together, these facts mean that there are very few measurements of atom-electron cross--sections available from laboratory experiments. Indeed, most are determined purely from theoretical work, see the volume edited by Brown & Lang (1988).

Chapter 3

Spectral decomposition by genetic forward modelling

This Chapter

In this chapter we take a brief side-step to look at a diagnostic (optimisation) method used extensively in the following chapters of this thesis; the Genetic Algorithm (GA). To demonstrate their operation and coding we apply a simple GA to the analysis of real and simulated line spectra (the GA applications presented in later chapters are merely extensions of this method). In particular, we show that this GA based technique experiences none of the user bias or systematic problems that arise when faced with poorly sampled or noisy data. An important feature of this technique is the ease with which rigid a priori constraints can be applied to the data. These constraints make the GA decomposition much more accurate and stable, especially at the limit of instrumental resolution, than decomposition algorithms commonly in use.

The launch of the SOlar and Heliospheric Observatory (SOHO) satellite discussed in Chapter 1, has renewed interest in the classification (Seely et al. 1997; Laming et al. 1997) and interpretation (Brekke et al. 1997; Judge et al. 1998) of high spectral resolution ultraviolet (UV) and extreme ultraviolet (EUV) emission spectra. The majority of these spectra come from the Solar Ultraviolet Measurement of Emitted Radiation (SUMER), and Coronal Diagnostic Spectrometer (CDS) instruments onboard SOHO (Wilhelm et al. 1995; Harrison et al. 1995).

A first step in the analysis of emission line spectra is to identify and measure properties of lines believed to be present. This is usually achieved by associating (subjectively) the observed spectral line profiles with ionic and atomic transitions of 'known' laboratory wavelengths.

From these possibly biased decompositions, physical models of the underlying plasma are sought using processes discussed in the following chapters of this thesis. In an effort to obtain the best possible scientific results from their spectra, the CDS and SUMER teams have set about ways to produce the most 'reliable' decomposition; see Brynildsen (1994) for more details.

Standard spectral decomposition techniques unfortunately prove to be unstable when presented with data of low signal to noise ratio, or data that is poorly sampled. In particular these instabilities cause subtle differences in the decomposition of each spectrum and can lead to significantly different physical interpretations. This has prompted us to search for a method that can provide spectroscopists with reliable decompositions of observed spectra that are as free as possible from subjective bias.

We use a heuristic approach to decomposition. We use a Genetic Algorithm (GA) to fit model line profiles, which for our purpose we chose to have Gaussian form, to provide a simple parameterisation of the spectrum under analysis. This approach exploits the stability and optimisation capabilities of natural selection (Darwin 1859). Sections 3.1.1 and 3.1.2 describe the basic GA formalism, and an introduction to our Gaussian fitting GA, hereafter referred to as Ga-GA.

The GA technique is applied under ideal conditions (to 'simple' noiseless test spectra) in Section 3.2.1. This first test also helps to highlight how well genetic operators are suited to this task. Section 3.2.2 gives a much more stringent test of the how a GA performs when fitting spectra containing unstructured random noise. Here, the GA's stability in the presence of random Gaussian noise is compared to that of standard profile fitting and optimisation algorithms. We show that these standard algorithms are blighted by possible user bias which is *not* present with the GA technique. To aid further comparison of our GA technique to standard analysis algorithms we have constructed model spectra with realistic noise and continuum/background levels. The results are discussed in Section 3.2.3.

The ability of the GA approach is given a final test in Section 3.3 on quiet Sun SUMER spectra. There we compare our results with those obtained from an analytical decomposition performed by Judge et al. (1998). We note that their technique used additional information not available to the GA.

Although much emphasis must be placed on the fact a GA requires minimal user input, in certain circumstances user input can prove useful, such as cases where relative wavelengths and intensities are well known from atomic physics. Such additional constraints can (almost

trivially) be 'hard-wired' into the algorithm. Section 3.3.1 highlights the possibilities of applying rigid *a priori* constraints to the observed spectrum.

3.1 Motivation and method

Prior to the launch of SOHO, a study was undertaken Brynildsen (1994) to identify the 'best' profile fitting package for the CDS and SUMER instruments discussed previously. The study compared various algorithms for fitting Gaussian profiles, or combinations thereof.

The common denominator linking all of the profile fitting algorithms studied by Brynildsen (CURVEFIT - from the Interactive Data Language (IDL) userlib, and AMOEBA - A "downhill" SIMPLEX algorithm from Press et al. 1992, and others) is the need for user input regarding starting points for *each* parameter in the search. This potential source of user bias, and the reduced quality (in terms of fit to the data) of the parameterisation form the principal motivation for this chapter, and indeed we show that they are not present using a GA technique beyond the absolute minimum requirement of supplying a 'line list' of lines to be identified.

Using a GA for this profile fitting problem can have many advantages not available to the user of predictive line fitting algorithms. Considering one of the many advantages noted in Charbonneau (1995), a GA is not de-stabilised by noise in the data; it will merely attempt to achieve its goal, locating the 'best' profile. The GA will attain this goal, the introduction of data noise will merely affect the convergence time of the algorithm.

We present a 'simple' GA, called Ga-GA, which we show to be stable against reasonable noise levels and to have no source of possible user bias. The following sections discuss its performance in detail.

3.1.1 Overview of a simple Genetic Algorithm

Genetic Algorithms are inspired by the mechanism of natural selection and basic genetic operators, occuring naturally in biological systems, see Holland (1962). Consider a typical numerical optimisation task, where a parametric model is to be fit to data in a manner that maximises the closeness of fit, or *fitness* (as measured, for example, by a χ^2 statistical estimator). A genetic algorithm is an iterative scheme that operates on a *population* of trial solutions to the problem in the following way :

1. Construct an initial population using *random* values for the model parameters, and evaluate their fitness.

2. Select a subset of the fitter individuals, and breed them to produce a new population.

3. Evaluate the fitness of each individual in the new population.

4. Replace the old population with the new one

5. Check whether the fitness has reached some pre-defined tolerance, or the number of iterations (or *generations*) has reached its maximum; if not return to step 2.

GAs carry out a form of forward modelling, by performing a heuristic search of the problem's parameter space. What distinguishes a GA from other forward modelling methods (such as Monte Carlo simulation) is primarily the way in which new trial solutions are constructed from the current population of trial solutions (cf. step 2 above).

At the most basic level a GA can be viewed as a processor of a set of strings, each encoding a particular version of the model being optimised. A subset of the fitter individuals of the current population are selected and paired, and the defining strings of each such pair are subjected to the action of two genetic operators: *cross-over* and *mutation*. The cross-over operation involves dissection of the two parent strings at a randomly chosen point along the string, followed by the interchange of the dissected components. In this way two new strings are produced from two parent strings (see figure 3.1). The second operator, *mutation*, involves the replacement of a few randomly selected digit in the two strings produced by the cross-over operation with randomly generated digit values. Its primary purpose is to maintain a suitable level of variation in the population, which is essential for selection to operate. The combination of stochastic genetic operators with fitness-based selection yields a powerful search algorithm that can move away from secondary extrema and locate the *global* extremum in parameter space (see, e.g., Goldberg 1989; Davis 1991; Charbonneau 1995; Mitchell 1996)

In this chapter we are using a GA version which implements a scheme involving a *variable* mutation rate, i.e. as the population becomes more degenerate (little variation) the probability of a mutation taking place is correspondingly increased, and makes use of *elitism*, the best individual in the old generation is *not* replaced unless there is a fitter one in the new generation. The selection of individuals in the breeding operator is carried out using a *roulette-wheel* algorithm (see Davis 1991, Ch. 1), meaning that individuals with higher fitness

are associated with sectors of correspondingly large angle on the roulette wheel. This roulette wheel, when 'spun', ensures that although all individuals are capable of breeding, the fitter individuals have a slightly higher probability of being selected.

In many ways our GA resembles that of Charbonneau (1995), but it also contains some features of the GA presented in Diver & Ireland (1997). Indeed, in the cases presented in Section 3.3 we have employed a variation on the algorithm PIKAIA, presented in Charbonneau (1995), to maximise speed and accuracy.

3.1.2 Fitness evaluation

Isolation of particular features (e.g. line width and absolute intensity) in an emission line spectrum made up of N lines is a procedure used by many standard fitting algorithms, with many using line identification as their primary 'search' (cf. the user input given to the algorithms mentioned above). On acquiring the line position they sequentially alter the amplitude or the $\frac{1}{e}$ width of the Gaussian profile(s) to achieve the 'best' fit to the target. However, since the observed emission line spectra can and do, contain a large number of profiles, it is possible to adopt a method which solves for all lines simultaneously (see e.g., Diver 1995; Diver & Ireland 1997).

When Ga-GA 'recognises' spectral features, i.e. one of the Gaussian describing parameters or an entire profile, the corresponding final solution will be a better representation of the target and will result in that string of parameters being given a higher fitness. Since Ga-GA uses the mechanics of natural selection, a genotype with a higher fitness value will be prevalent in the current and future generations until replaced by a 'fitter' individual.

Ga-GA uses parameter strings of length $3 \times N$, where N is the estimated number of Gaussian profiles in the line spectrum to be analysed, and three because it requires three

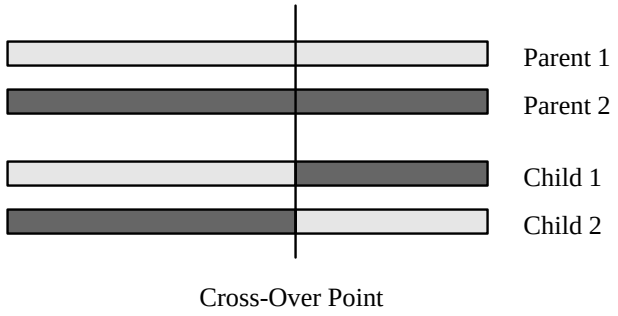

Figure 3.1: A pictorial explanation of the main GA breeding operator, *cross-over*.

parameters to describe a general Gaussian profile. These parameters are absolute position in wavelength, at channel (X), amplitude (A), and the Gaussian's $\frac{1}{e}$ value (W) and are encoded as a string in the following order :

$$[X_1, A_1, W_1, \ldots\ldots, X_N, A_N, W_N]$$

A string of the form above defines a sequence of N Gaussian profiles that defines a synthetic spectrum, this computed profile is an individual's phenotype. It is this phenotype profile that is retained for comparison to the observed spectrum. Phenotype profiles are calculated using the standard pointwise Gaussian formula, i.e. for a particular channel x, usually associated with wavelength, in Gaussian i (G_i) :

$$G_i(x) = A_i \exp\left(\frac{-(x - X_i)^2}{W_i^2}\right) \tag{3.1}$$

The N Gaussian profiles derived from a particular genotype string are summed to form the 'unique' phenotypic profile for genotype j, $P(\underline{x})_j$ (with $\underline{x}$ meaning for all channels x). $P(\underline{x})_j$ is given by :

$$P(\underline{x})_j = \sum_{i=1}^{N} G_i(x) \quad \forall x \tag{3.2}$$

Only once $P(\underline{x})_j$ has been computed do we calculate an *error* measure between it and the target. The error measure of a particular genotype ($E(\underline{x})_j$) depends on several factors; the square pointwise difference of the target and the corresponding phenotype ($C(\underline{x})$, and $P(\underline{x})_j$), the number of parameters in the calculation ($3 \times N$), the number of points summed over (Ndata) and an estimate of the noise level in the data ($\sigma_{data}(\underline{x})$). Thus, $E(\underline{x})_j$ (effectively a normalised χ^2 measure) is given by :

$$E(\underline{x})_j = \frac{1}{(Ndata - 3N)} \sum_x \left(\frac{(C(\underline{x}) - P(\underline{x})_j)}{\sigma_{data}(\underline{x})}\right)^2 \tag{3.3}$$

with $E(\underline{x})_j \sim 1$ indicating a 'good' fit.

This measure is used to evaluate the fitness of each genotype. It is the fitness value that is used to rank all the genotypes in a particular population into ascending order and to 'weight' the roulette wheel of Section 3.1.1.

3.2 Results

This section details the results of Ga-GA applied to simulated target data sets which have a known level of noise added. Section 3.2.1 discusses the performance of Ga-GA in the absence

of data noise (except for very small numerical rounding errors). Sections 3.2.2 and 3.2.3 provide ideal circumstances to test the performance of Ga-GA, against that of the two standard algorithms mentioned earlier; CURVEFIT and AMOEBA, for data with a realistic noise level and with a noisy background present (Sections 3.2.2 and 3.2.3 respectively). Section 3.2.3 will also show the ease with which additional spectral features may be incorporated into the analysis.

3.2.1 Application to noiseless target spectra

We use Ga-GA to analyse three noiseless targets, i.e. we replace $\sigma_{data}(\underline{x})$ by 1 in equation (3.3), each corresponding to a different Gaussian configuration. The three test targets are: 1) A single 'wide' Gaussian with the target genotype given by three parameters, $[\,X\,A\,W\,] = [\,50\,100\,20\,]$. 2) Two 'joined' Gaussians corresponding to the six parameter genotype $[\,40\,100\,20\;\;80\,90\,15\,]$, and 3) a more complex five Gaussian configuration with the fifteen parameter target genotype given by $[\,10\,30\,5\;\;22\,60\,1\;\;26\,40\,3\;\;43\,70\,5\;\;55\,60\,5\,]$.

Each case was analysed ten times (to allow performance statistics to be compiled), each run with a different initial population, for a fixed number of generations. It is also possible to configure Ga-GA to run until it achieves a fixed $E(\underline{x})$ although for certain types of analysis this method is unfavourable (Charbonneau & Knapp 1996). The number of generations used in each case is different however, and varies with the increase in complexity of the target solution. Therefore target 3 typically requires a 1200 generation run, which is considerably more than the 200 and 500 generation runs required for targets 1 and 2 respectively.

The returned parameterisation of each target is given in Table 3.1. The subscript T quantities (e.g. X_T) are the target parameters and the subscript G quantities (e.g. X_G) are the corresponding mean values returned by Ga-GA after multiple fixed generation runs. It is clear from the results presented in Table 3.1 that Ga-GA obtains a *very* good representation of each target (within the errors). The errors in the parameters are global error estimates and are calculated in a Monte Carlo fashion, i.e. we perform multiple runs of Ga-GA each with a different initial population, this is achieved by initializing the random number generator with a different seed (Charbonneau & Knapp 1996). This Monte Carlo approach 'forces' Ga-GA to search the parameter space from a different starting point each time. This will also allow the calculation of 'mean' values for each of the parameters.

Figure 3.2 shows a plot of target 1 (solid line) and the profile derived from the 'fittest' genotype ($\triangle$) after only 200 generations with the $E(\underline{x}) = 2.476 \times 10^{-4}$. Similarly, figure 3.3

shows the profile constructed from the fittest genotype, $E(\underline{x}) = 3.296 \times 10^{-3}$, for the double Gaussian configuration of target 2. Figure 3.4 demonstrates Ga-GA's handling of the more complex case 3, resulting in $E(\underline{x}) = 1.984 \times 10^{-4}$ of the fittest genotype after 1200 generations. For these test cases final values of $E(\underline{x})$, if we doubled the number of generations, will be limited by numerical precision and would possibly attain no better values than those given and it must be emphasised that these results are for *one* particular run of Ga-GA from the ensemble of 10 runs.

We show, in figure 3.5, the decrease in $E(\underline{x})$ with generation number for the full ensemble of runs (indicating the mean $E(\underline{x})$ (solid line), extrema (dashed line) and median (dotted line) for each generation step) for each of the test cases above. These plots demonstrate the power of Genetic Algorithms as optimisation tools. The steplike structure is clearly visible in all three plots, although to a much greater extent in the uppermost plot. Such steps occur when Ga-GA suddenly obtains a new 'fitter' value for one (or more) parameter(s), the long flat 'plateaus' are points where the current 'best' in the population hasn't changed or when the population is largely degenerate, i.e. all the individuals have very similar genotypes. These mutation jumps will occur because the mutation rate has been allowed to increase, and will

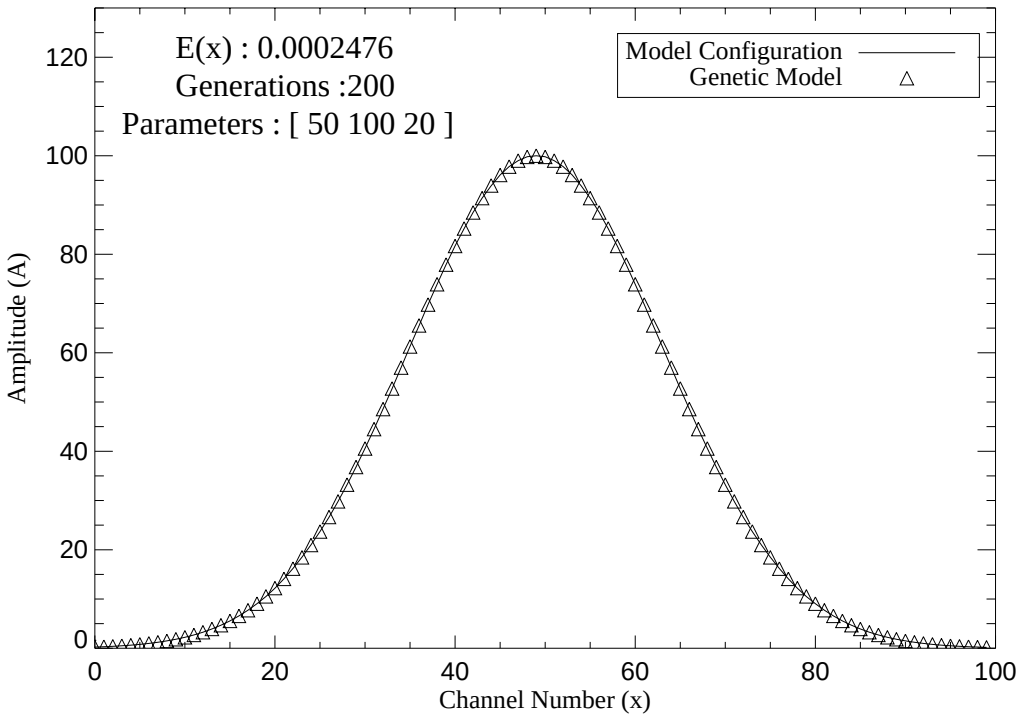

Figure 3.2: Test run for Ga-GA, taken from the ensemble of ten runs, for the *noiseless* single Gaussian target (solid line) of Case 1 and the profile modelled by Ga-GA (△).

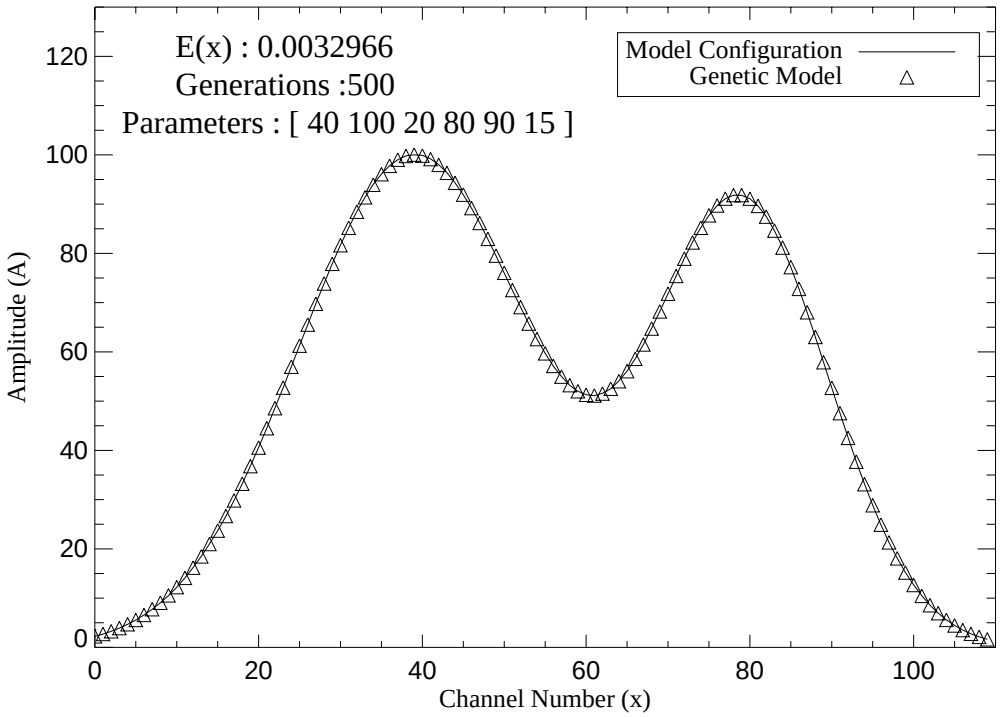

Figure 3.3: Test run for Ga-GA, taken from the ensemble of ten runs, for the *noiseless* double Gaussian target (solid line) of Case 2 and the profile modelled by Ga-GA (△).

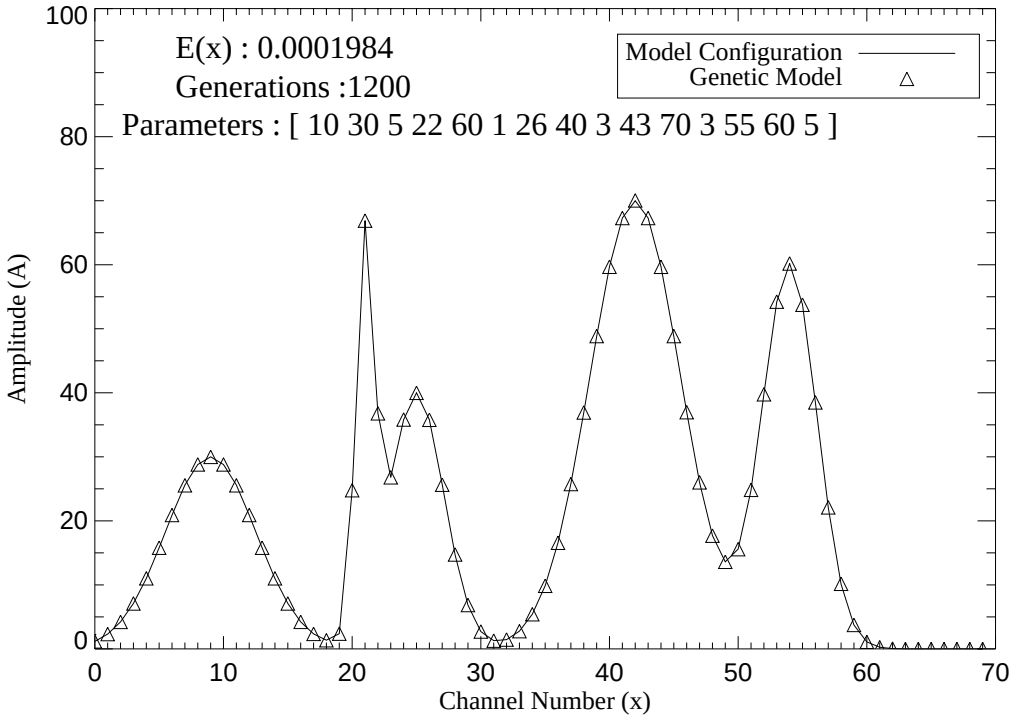

Figure 3.4: Test run for Ga-GA, taken from the ensemble of ten runs, for the *noiseless* five Gaussian target (solid line) of Case 3 and the profile modelled by Ga-GA (△).

Table 3.1: Results for cases 1), 2) and 3) described above. Subscript T quantities indicate target parameters, and subscript G quantities are the mean after multiple evolutionary runs. Similarly, the values of $\langle E(\underline{x})\rangle$ are the final mean values of $E(\underline{x})$. The errors for each parameter are calculated as the means of the ten run ensemble.

X_T	A_T	W_T	$X_G \pm \delta X_G$	$A_G \pm \delta A_G$	$W_G \pm \delta W_G$
	Case 1.	$\langle E(\underline{x})\rangle$	5.226×10^{-4}	200 gens.	
50.00	100.0	20.00	50.000 ± 0.000	100.002 ± 0.003	20.002 ± 0.001
	Case 2.	$\langle E(\underline{x})\rangle$	3.779×10^{-3}	500 gens.	
40.00	100.0	20.00	40.002 ± 0.002	100.007 ± 0.007	20.004 ± 0.003
80.00	90.00	15.00	79.997 ± 0.002	89.998 ± 0.003	14.999 ± 0.002
	Case 3.	$\langle E(\underline{x})\rangle$	7.623×10^{-4}	1200 gens.	
10.00	30.00	5.000	9.998 ± 0.001	30.003 ± 0.019	4.997 ± 0.004
22.00	60.00	1.000	21.997 ± 0.001	59.661 ± 0.181	0.995 ± 0.002
26.00	40.00	3.000	25.983 ± 0.007	39.867 ± 0.068	3.002 ± 0.006
43.00	70.00	3.000	43.000 ± 0.000	69.951 ± 0.028	3.001 ± 0.001
55.00	60.00	5.000	54.999 ± 0.001	59.964 ± 0.014	5.003 ± 0.001

thus introduce new genetic material at a higher frequency.

Figure 3.5 also justifies our earlier claim that more complex targets (more parameters) require a greater number generations in the run. As with any optimisation method the plots show how the gradient of $E(\underline{x})$ lessens with the increase in the number of parameters in the genotype, the increase in the number of generations required for a GA to evolve an acceptable solution increases with the dimension, D, of the search space; typically it does so in a manner that is highly problem dependent, but often ends up as being a low (order unity) power of N. So such convergence plots provide evidence to suggest that we have not yet evolved a 'perfect' match for the target. This may be estimated by looking at the gradient of the plot at the end of its evolutionary run. The center and bottom plots in figure 3.5 show that the evolutionary process may not be finished.

3.2.2 Application to a 'noisy' target spectrum

Reliable analysis of a 'noisy' target must be the benchmark for any spectral decomposition technique. We therefore compare the performance of Ga-GA to that of the AMOEBA and

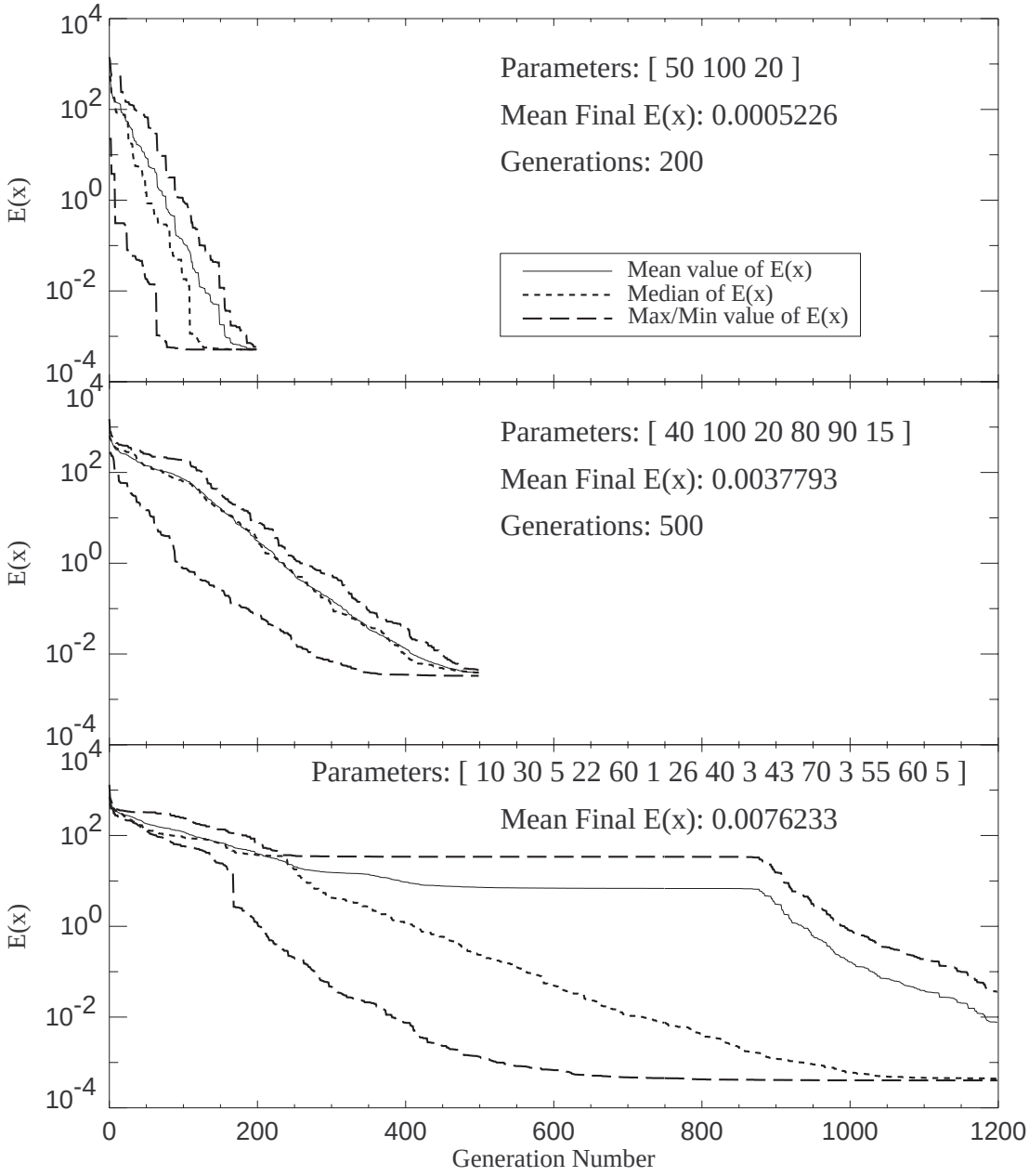

Figure 3.5: Convergence of $E(\underline{x})$ against generation number for each of the three cases in Section 3.2.1. Top panel : Case 1 (single Gaussian), Middle panel : Case 2 (two Gaussians) and Bottom panel : Case 3 (five Gaussians). For each generation step the mean $E(\underline{x})$ (solid line), median (dotted line) and extrema (dashed lines), for the ten run ensemble, are indicated. It is clear that, when a relatively 'poor' parameterisation is present, the difference between the median and mean of $E(\underline{x})$ is demonstratably effected, this effect is evident in the top and bottom panels.

CURVEFIT algorithms in decomposing a 'noisy' five Gaussian target, again with Ga-GA results the mean of ten runs. The target is generated by the same fifteen parameter genotype as case 3 of Section 3.2.1 ([10 30 5 22 60 1 26 40 3 43 70 3 55 60 5]) to which we now add 15% 'random' noise. The noise is set to be normally distributed about the data with an r.m.s. amplitude of 15%, so $\sigma_{data}(\underline{x}) = 0.15\, C(\underline{x})$ in equation (3.3).

The results of the calculations for each algorithm[1] are shown in Table 3.2 where Ga-GA achieves the lowest $E(\underline{x})$ (1.889), by a factor of six from CURVEFIT (12.961) and by a factor of about ten from AMOEBA (18.626). It must be noted that all produce 'good' parameterisations of the spectrum given the severe noise present, but bear in mind that the latter two algorithms are practically given the target parameters as a startpoint, and are hence heavily influenced by the user. This is definitely not the case with Ga-GA. CURVEFIT and AMOEBA also exhibit another behavioural pattern not observed with Ga-GA; they will occasionally become 'stuck' at points in the solution space where hope of convergence to the target is lost[2]. This does not happen in every run, but indicates to the user that a single run using either method is not enough to guarantee a reliable parameterisation.

Figure 3.6 shows the results of Ga-GA ($*$), CURVEFIT ($+$) and AMOEBA ($\diamondsuit$) operating on the fifteen parameter, five Gaussian target. The profile shown for Ga-GA, as in Section 3.2.1, is the 'fittest' phenotype from the ten different runs. It is clear from the results in Table 3.2, and the plots in figure 3.6 that the sharp features of Gaussian two (at a possible limit of resolution) present CURVEFIT and AMOEBA with a very awkward test. Indeed, by inspection of the errors quoted in Table 3.2 it is possible to see the feature(s) that Ga-GA finds most awkward to 'identify', these are the amplitudes A_2, A_3 and A_4.

3.2.3 Application to a target with a background level

We now consider the case where the target has a considerable background level. A GA approach makes inclusion of such a background, or continuum, extremely simple. To show this, consider a parameterisation of the background by addition of a quadratic of order n, an example for $n = 2$ is given in equation (3.4). As an example, consider a new three Gaussian configuration [10 90 6 50 70 3 80 40 4] with 5 % noise ($\sigma_{data}(\underline{x}) = 0.05\, C(\underline{x})$) and background; the alteration to the fitness evaluation routine is minimal. We add the quadratic form to the

[1] It should be noted that CURVEFIT and AMOEBA were initialised with a guess of each parameter that is within ± 2 the target parameter value.

[2] The interested reader is directed to Charbonneau & Knapp (1996) for a discussion of this effect.

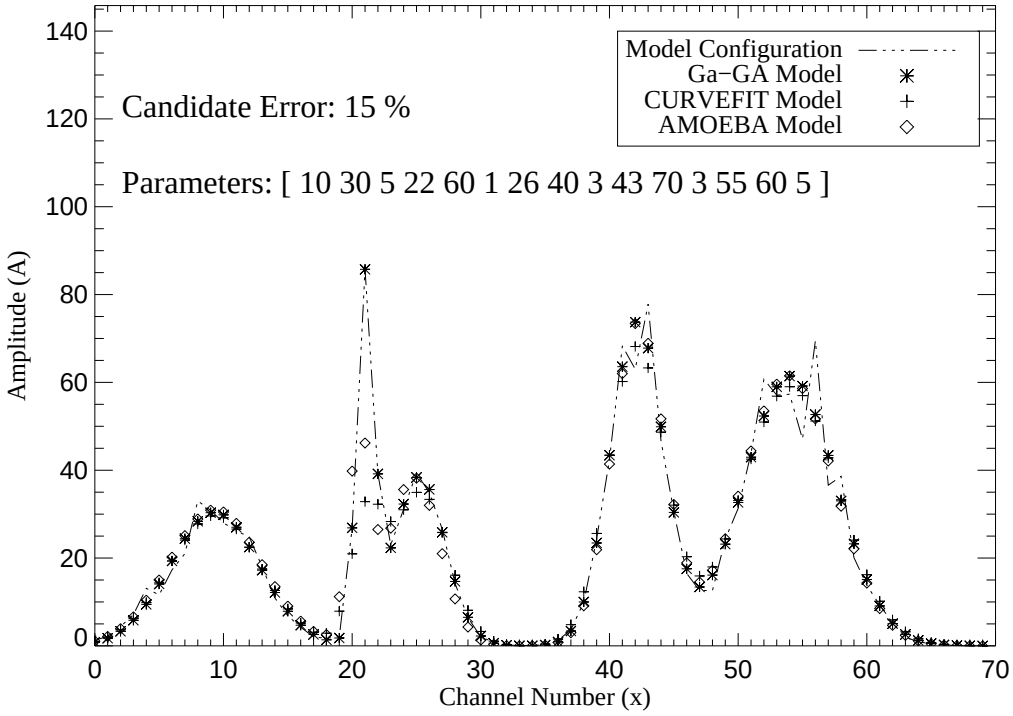

Figure 3.6: Performance comparison plot between Ga-GA, AMOEBA and CURVEFIT. They are compared using the target of Section 3.2.2 with 15% added random noise. See also Table 3.2.

Table 3.2: Details of the target parameters(P_T), genetically modelled solution returned by Ga-GA and the deterministic routines for the fifteen parameter configuration with 15% normally distributed random noise. Ga-GA results and CPU times (T_{CPU}) are the mean of an ensemble of ten runs. The CPU times are normalised to the CPU time of a CURVEFIT run.

P	P_T	AMOEBA	CURVEFIT	Ga-GA
X_1	10.00	10.340	9.317	10.305 ± 0.001
A_1	30.00	31.002	29.700	30.433 ± 0.011
W_1	5.000	5.101	5.062	4.908 ± 0.003
X_2	22.00	21.552	21.092	22.024 ± 0.001
A_2	60.00	45.021	27.985	81.160 ± 0.073
W_2	1.000	1.253	1.744	0.947 ± 0.001
X_3	26.00	25.790	25.305	26.187 ± 0.009
A_3	40.00	38.408	35.121	38.506 ± 0.031
W_3	3.000	2.843	3.051	2.856 ± 0.006
X_4	43.00	43.210	42.100	43.122 ± 0.001
A_4	70.00	73.502	67.914	73.611 ± 0.017
W_4	3.000	2.915	3.138	2.919 ± 0.001
X_5	55.00	54.887	54.016	55.018 ± 0.001
A_5	60.00	61.449	59.015	61.433 ± 0.001
W_5	5.000	5.055	5.265	5.050 ± 0.001
T_{CPU}		114.125	1.000	109.312
$E(\underline{x})$		18.626	12.961	1.889

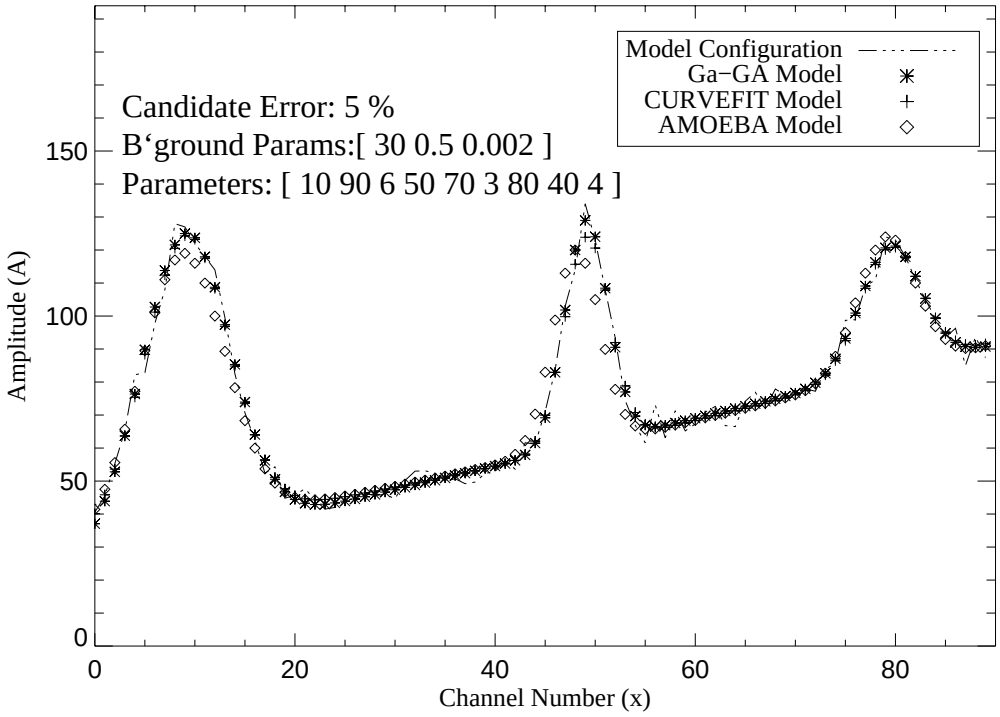

Figure 3.7: Plot of the three Gaussian configuration [10 90 6 50 70 3 80 40 4] and the background parameters, $a = 30.0, b = 0.5$, and $c = 0.002$ with a 5% random noise level. See also Table 3.3.

standard phenotype calculation of equation (3.1), which then becomes:

$$P(\underline{x})_j = a + bx + cx^2 + \sum_{i=1}^{N} G_i(x) \tag{3.4}$$

where a, b, and c are taken from the adapted genotype by adding $[\,a\,b\,c\,]$ to the Gaussian description parameters. To generate the target the background parameters are assigned the values $a = 30.0, b = 0.5$ and $c = 0.002$.

A plot of the target solution (broken line) and the best phenotype ($*$) is shown in figure 3.7. The figure also shows the profile returned by CURVEFIT ($+$) and that returned by AMOEBA ($\diamondsuit$). Ga-GA's estimate of the background parameters are $a = 29.243, b = 0.554$ and $c = 0.002$ (with respective errors given below). Ga-GA results were returned after 1000 generations and the mean final $E(\underline{x})$ was 0.8664, with CURVEFIT giving a statistically equivalent fit (0.8600) and AMOEBA by a factor of two (2.000). The full results of the parameterisation for all three algorithms are given in Table 3.3.

Table 3.3: Results from Section 3.2.3 for a target ($P(T)$) with fixed background level and 5% normally distributed random noise. Again, Ga-GA results and CPU times (T_{CPU}) are the mean of an ensemble of ten runs. CPU times are normalised to that of a CURVEFIT run.

P	P_T	AMOEBA	CURVEFIT	Ga-GA
X_1	10.00	9.810	9.172	10.131 ± 0.017
A_1	90.00	82.404	88.062	90.926 ± 0.529
W_1	6.000	6.020	5.980	6.045 ± 0.056
X_2	50.00	49.100	49.160	50.101 ± 0.001
A_2	70.00	60.001	63.526	68.474 ± 0.121
W_2	3.000	3.312	3.238	3.059 ± 0.009
X_3	80.00	80.103	79.469	80.429 ± 0.001
A_3	40.00	41.261	37.544	38.340 ± 0.077
W_3	4.000	4.121	4.303	4.357 ± 0.015
a	30.00	31.180	31.740	29.243 ± 0.964
b	0.500	0.501	0.486	0.554 ± 0.037
c	0.002	0.002	0.002	0.002 ± 0.000
T_{CPU}		80.134	1.000	75.321
$E(\underline{x})$		2.000	0.8600	0.8664

3.3 Analysis of a quiet Sun SUMER spectrum

To test Ga-GA on real data we chose to analyse a spectral region in the SUMER wavelength range that is known to suffer from blending problems, both between spectra of different optical orders as well as just wavelength coincidences. Those problems resulting from blends between lines that happen to overlap in the first and second grating orders can be decomposed experimentally, and thus serve as a limited check on the GA approach.

The dataset analysed here was obtained on October 26th 1996, with the 1×300 arcsecond slit crossing the north polar limb, using SUMER's B detector. Data were acquired in the 1400 Å spectral region, containing strong lines of Si IV, O IV, and O III (in second order), as well as other weaker lines.

The observing sequence was designed to obtain data between 1399 and 1408 Å (and in the second order spectrum with wavelengths at half of this range) on both the bare and KBr coated part of the detector, sequentially. The exposure time on the KBr part was 180 seconds, and 360 seconds on the bare part. The bare and KBr regions of the detector have very different sensitivities to first and second order spectra. Assuming that the spectra did not change significantly between the bare and KBr exposures, the different count rates acquired on the two regions allow one to decompose the spectrum analytically into first and second order components, I_1 and I_2 through the following equations

$$Cts(KBr) = k_1 I_1 + k_2 I_2 \tag{3.5}$$

$$Cts(bare) = b_1 I_1 + b_2 I_2 \tag{3.6}$$

where $Cts(KBr)$ and $Cts(bare)$ refer to the count rates per pixel per second on the KBr and bare parts of the detector, I_1, I_2 are intensities of the first and second order spectra, and k_1, k_2, b_1, b_2, are (known) instrument sensitivities defined through these equations. Figure 3.8, top panel, shows $Cts(KBr)$ and its components, $k_1 I_1$ and $k_2 I_2$. Values for I_1 and I_2 were obtained using measurements of $Cts(KBr)$, $Cts(bare)$ and instrumental sensitivities discussed by Judge et al. (1998). Figure 3.8 also shows $Cts(bare)$ and its components, in the bottom panel. In each case the count rates are averaged over 300 spatial pixels, including the solar limb, and time during the exposures.

Shown in the top panel of figure 3.9 is a decomposition performed using Ga-GA based *only* upon the $Cts(KBr)$ spectrum shown in the upper panel of figure 3.8. This is simply a 'blind' fit, using *no* prior information about the spectrum, except that we expect between 16

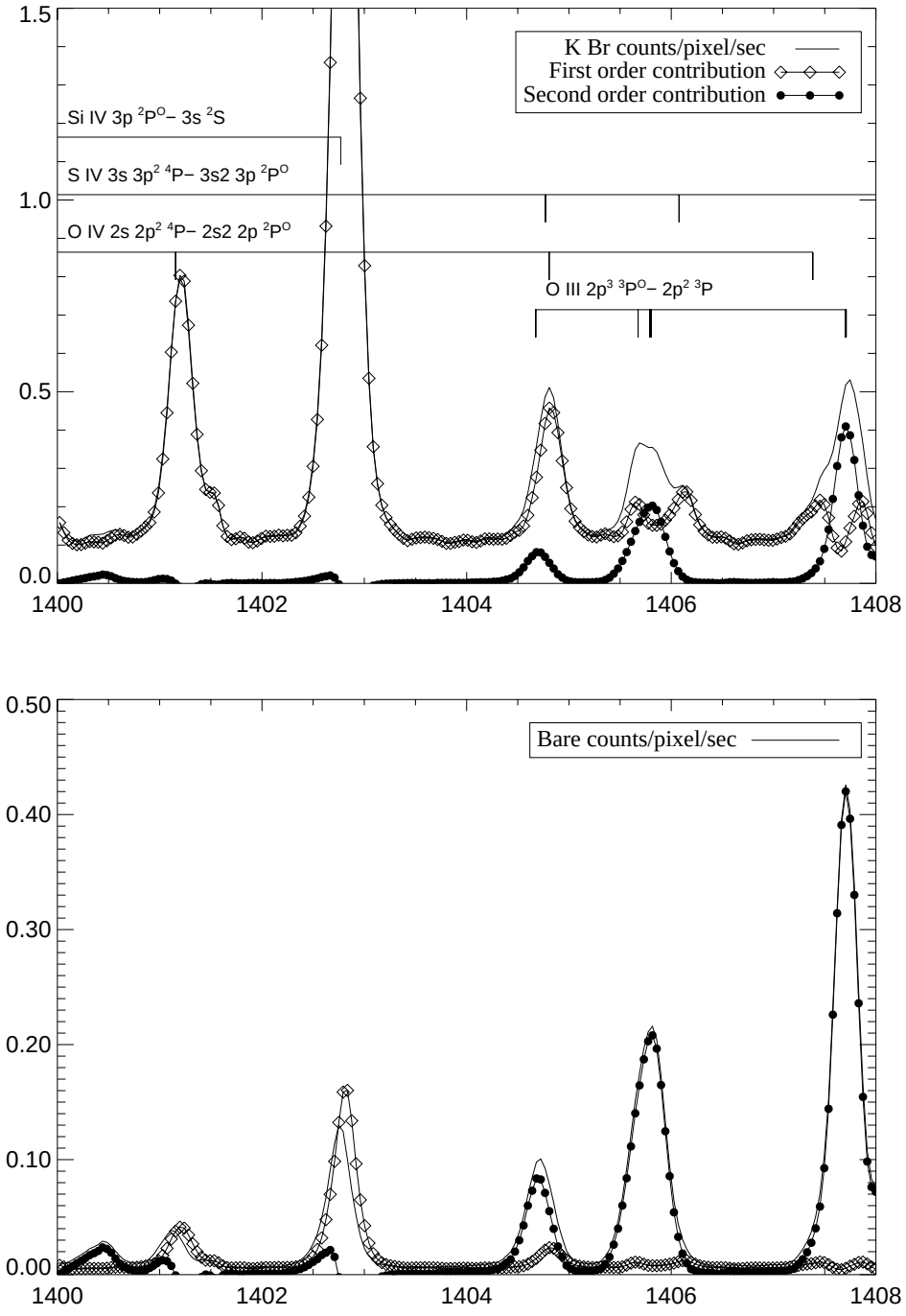

Figure 3.8: The 1400 Å region of the solar spectrum as measured using the SUMER instrument (see text for details). The top panel shows the average spectrum, in counts/pixel/second, recorded on the KBr region of the detector. Positions of known strong lines are marked- the positions of lines of O III are marked assuming that they are formed in the second order. The bottom panel shows the same thing, but recorded on the bare part of the detector. The lines plotted with symbols show the spectral decomposition into first and second order lines using the known sensitivities from SUMER.

and 20 Gaussians to be present with on constant background. Such 'blind' fits show that we can obtain a reliable decomposition of the entire spectrum. An example where a 'blind' run is significantly better than one where *a priori* knowledge is used to aid in the decomposition is given below (see Table 3.4).

3.3.1 Using Additional Knowledge

Usually, extra information about the spectrum is known, and it may be needed for some cases. This information can be 'hard-wired' into Ga-GA easily. For example, we could demand that the spectral decomposition must not contain spectral detail narrower than the instrumental width (σ_{inst}). Or, we could specify that relative positions (or intensities) of lines from the same ion, known to great accuracy from laboratory measurement, be fixed to certain values. Such constraints can be incorporated into the GA through a simple modification of the fitness evaluation, equation (3.3). For such an example we might use:

$$E(\underline{x}) = \chi^2 + C_i H^2(W_i, \sigma_{inst}) + D_{ij}\left((X_i - X_j) - (X_i^{lab} - X_j^{lab})\right)^2 + \ldots \qquad (3.7)$$

where we introduce the additional constants C_i and D_{ij} to control the 'trade-off' between χ^2 and the newly incorporated information, and where $H(W_i, \sigma_{inst})$ will weight the optimisation against features narrower than σ_{inst}. A future version of Ga-GA may take advantage of this additional information to act as desktop on-line plasma analysis package. Recall however, that the number of parameters in the calculation effects the rate of convergence (Section 3.2.1 and Section 3.2.2).

The lower panel of figure 3.9 shows the results of a Ga-GA decomposition where we have included a line list of all the lines marked in upper panel of figure 3.8, the implementation of this is discussed below. The 'fixed' wavelength decomposition[3] (see results in Table 3.4) tells us additional information about the spectrum; there is an average redshift of 0.070 Å of the lines in the list from their reference position. This corresponds to a velocity of around 10 km/s. The comparison of the contributions between first and second order lines in the 1404 - 1408 Å region shows that Ga-GA can successfully decompose a real, convoluted spectrum, into meaningful components.

[3]The profiles computed are allowed to deviate from the reference wavelength by, at most 0.1 Å.

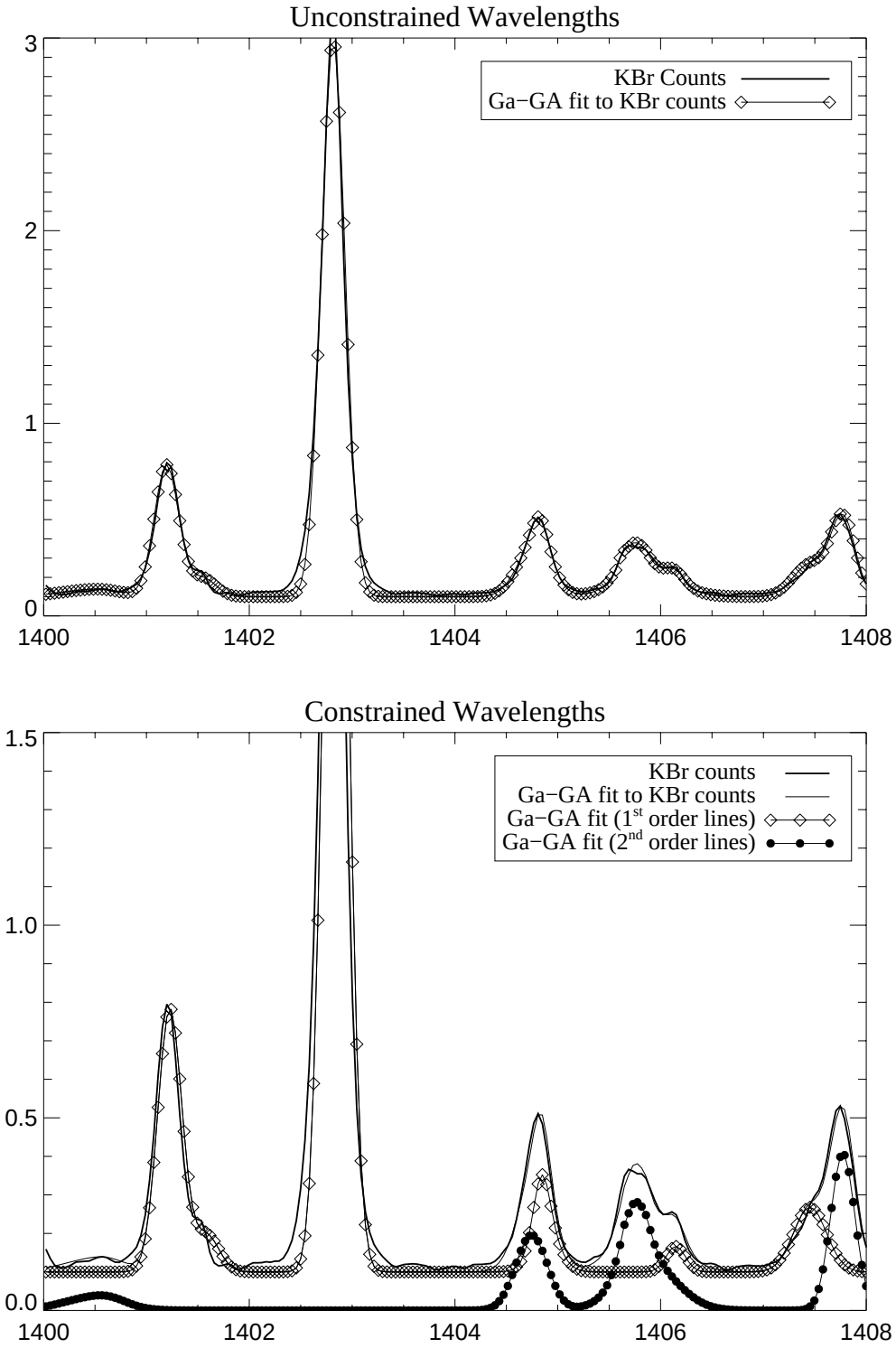

Figure 3.9: Comparison between Ga-GA decomposition and the analytic decomposition of the SUMER spectrum in figure 3.8. The top panel shows the decomposition from the Ga-GA algorithm using only the KBr data from the top panel of figure 3.8. The bottom panel shows the decomposition from a *single* run of Ga-GA using constrained wavelengths in the fitness calculation. See Table 3.4 for the details of the runs with constrained wavelengths.

Table 3.4: This table contains the results of Ga-GA analysing the SUMER spectrum of figure 3.8 where the wavelengths, $\langle\lambda_G\rangle$ (Å), intensities, $\langle I_G\rangle$, and widths $\langle W_G\rangle$ (Å) are the mean values of a ten run ensemble. † indicates that, in this wavelength range, a line of Ar VIII at $\lambda = 700.245$ Å (in second order) dominates the emission, as is clear from inspection of images shown by Judge et al. (1998) but this was not given in the line list. This line *was* detected in the 'blind' decomposition of Section 3.3 ($\lambda_G = 1400.558$Å, $I_G = 0.030$ and $W_G = 0.151$Å) with correspondingly different measurements for the two lines of S III. This result illustrates that *a priori* information (in this case, the line list), must be correct or erroneous results will occur. Mean standard deviations in $\langle I_G\rangle$ and $\langle W_G\rangle$ are 0.002 and 0.001 respectively.

Ion	Order	λ_{ref}	$\langle\lambda_G\rangle$	$\langle I_G\rangle$	$\langle W_G\rangle$
S III†	2	1400.374	1400.449	0.019	0.406
S III†	2	1400.573	1400.648	0.027	0.408
O III	1	1401.157	1401.232	0.692	0.162
S I	1	1401.514	1401.589	0.110	0.145
S IV	1	1402.770	1402.845	2.875	0.164
S IV	1	1404.771	1404.846	0.368	0.195
O IV	1	1404.806	1404.881	0.018	0.411
S III	2	1405.566	1405.641	0.044	0.094
S III	2	1405.643	1405.718	0.108	0.230
O III	2	1405.676	1405.751	0.092	0.002
O III	2	1405.791	1405.874	0.014	0.060
S IV	1	1406.076	1406.151	0.086	0.139
O IV	1	1407.382	1407.457	0.138	0.308
O III	2	1407.701	1407.776	0.091	0.122
O III	2	1407.709	1407.784	0.270	0.207

3.4 Discussion

We have presented a heuristic search algorithm for the detection and analysis of spectral lines, which is free of operator bias and robust against poor or noisy data. Data are fitted simultaneously, and not sequentially, therefore limiting the propagation of systematic errors through the procedure. Coding is simple to write and easy to use, needing minimal operator input. However, the simplicity of the GA used here places limitations on the amount of information that can be extracted from spectra. Although there is *no* practical limit to the number of parameters used in the genetic decomposition, the efficiency with which the one point cross-over operator 'explores' the solution space decreases as the number of parameters increases. However, such a problem can be countered simply by using a multiple point cross-over operator (see discussion in Goldberg 1989). Such adaptations are simple to make in any GA code.

In cases where data is more poorly sampled or noisier than those examined here, convergence times may become longer than the few minutes or so typical of the examples shown. It is clear from the CPU times (T_{CPU}) given in Tables 3.1 and 3.2 that although Ga-GA is not as 'fast' as CURVEFIT we can see that the user must compromise between run time and the degree of accuracy required since Ga-GA has clearly demonstrated its usefulness in the presence of quite severe noise. Presumably there is also a trade-off between poorer sampling (i.e. fewer points) saving on floating point operations, and noisier data leading to many more fitting attempts. Monitoring the convergence of the GA in the cases examined here indicates that it is adept at rapidly fitting the large scale spectral features, and progressively slower at smaller scales. This cascading nature is central to the operation of a GA, and underpins its stability in the face of noisy data (the noise being on the smallest scale is fitted last). Increasing the scale of the computation is straightforward since the generation of each child is an independent calculation (strictly, the generation of each pair of derived strings), and so the algorithm lends itself naturally to parallelisation. It is also clear that a GA routine like Ga-GA[4] could form part of a suite of line analysis codes, with the GA offering a best initial estimate of the profile for more conventional processing methods which require a 'good' initial guess.

[4] A version of the Fortran-77 Ga-GA code is given in Appendix A.1.

Chapter 4

New light on the solution of DEM inverse problems

This Chapter

Spectroscopic diagnosis of the temperature and density structure of hot optically thin plasmas from emission line intensities is usually described in two ways. The simplest approach, the 'line ratio' method, uses an observed ratio of emission line intensities to determine a 'spectroscopic mean value' of electron temperature $\langle T_e \rangle$ or electron density $\langle n_e \rangle$. The mean value is taken to be the homogeneous theoretical value of T_e or n_e which matches that ratio of observed line intensities. The line ratio method is stable, leading to well defined values of $\langle T_e \rangle$ or $\langle n_e \rangle$ for each line pair, but in the outer solar atmosphere (a highly inhomogeneous plasma) such mean values are hard to interpret since each line pair yields different mean parameter values. The more general 'differential emission measure' (DEM) method recognises that observed plasmas are better described by DEM distributions of temperature or density over the observed plasma volume, and poses the problem in the inverse form of deriving the DEM functions from the complete line set. It is well known that the DEM function is the solution to an inverse problem and can be treated as a function of T_e, n_e, or both. Derivation of DEM functions, while generally considered to more rigorous, is unstable to noise and errors in spectral and atomic data. This Chapter highlights work on the DEM inverse problems discussed in the previous chapters and presents a novel Genetic Algorithm based technique for circumventing the effects produced by systematic errors present in the atomic models.

Knowledge of the densities and temperatures of space plasmas is essential if we are to understand their most basic structure and transport processes in them. Without this knowledge, almost nothing can be said from data regarding the generation and transport of mass,

momentum and energy. Thus, since early in the era of space-borne spectroscopy, we have faced the task of inferring plasma electron densities, n_e, and temperatures, T_e, for hot solar and other astrophysical plasmas from optically thin emission line spectra (e.g., Gabriel & Jordan 1969; Munro et al. 1971; Gabriel & Jordan 1971; Dere & Mason 1981; Doschek 1987; Mason & Monsignori-Fossi 1994).

A fundamental property of hot solar plasmas is their basic inhomogeneity. This is obvious from direct images of the Sun's corona and transition region which show a wealth of fine scale structure down to the observable limits of resolution (e.g., see the recent book on the solar corona by Golub & Pasachoff 1997). It is confirmed by less direct spectroscopic work which reveals differing values of n_e, T_e for different line ratios (see, e.g., discussions in Doschek 1984 and Doschek 1987). Strong inhomogeneity is expected also from physical considerations (a particularly interesting perspective, addressing why the plasmas do not appear to be even more inhomogeneous than already observed, is given by Litwin & Rosner (1993)).

The emergent intensities of spectral lines at each 'point' in 2D images of optically thin plasmas are determined by integrals along the line of sight (i.e. the third dimension) through plasmas. There are two common approaches to inferring plasma properties from observed spectral line intensities. Consider the case in which a characteristic temperature of the electrons in the plasma is desired. The simplest approach, the 'line ratio' or 'spectroscopic mean' method, involves finding the single value of the electron temperature from a theoretical calculation of the ratio of carefully selected emission lines, that is in agreement with that single observed ratio. A "spectroscopic mean value" of the temperature $\langle T_e \rangle$ is derived for each line pair. If the plasma were truly isothermal, then the derived spectroscopic mean values for all line pairs would coincide with the actual plasma temperature, to within observational and theoretical errors. This approach was applied as early as 1941 to planetary nebulae by Menzel et al. (1941), and is reviewed by Gabriel & Jordan (1969) and Mason & Monsignori-Fossi (1994), see also Section 2.2.2.1. The other method is to recast the above mentioned line integrals into suitable form for 'inversion', in which one solves for a function, $\xi(T_e)$, which is a source term that describes the emissivity distribution of material as a function of temperature along the line of sight. $\xi(T_e)$ is called the 'Differential Emission Measure' (DEM) function, see Section 2.1.1.2. This gives a general characterisation of the distribution of the plasma with respect to temperature.

The integral equation formalism for temperature sensitive lines was first discussed by Pottasch (1964) and put on a rigorous mathematical basis for arbitrary geometry by Craig

& Brown (1976). The formulation was later extended to a DEM function $\zeta(n_e)$ differential in n_e for isothermal plasmas (Almleaky et al. 1989, and references therein). The concept was also generalised to the bivariate case of $\mu(n_e, T_e)$ by Jefferies et al. (1972a, b) although their definition contained an error corrected in the independent formulation by Brown et al. (1991). Formulation of how this general bivariate case could be cast as an inverse problem and in principle solved eluded these earlier authors and was finally established by Hubeny & Judge (1995) and elaborated by Judge et al. (1997).

Although more general than the line ratio method, it is well known that the DEM formulation is prone to errors in the solution arising from the ill-posed nature of the inverse operator - i.e. instability of the solution to errors in the spectral and atomic data (Craig & Brown 1976; Judge et al. 1997). This is intrinsic to the nature of the inverse problem, in which a continuous distribution function (or discretisation thereof) is sought from a finite number of data points and, furthermore, there is significant linear dependence between the line emissivities (or 'kernels') in the integral equations (see Sections 4.1.1 , 4.1.2 and 4.1.3 and discussion in the following chapters of this thesis). There is thus no single mathematical solution to the DEM problem, and the intrinsic ill-posedness (see Chapter 2) must be addressed from the outset, essentially by smoothing the desired DEM function so that, in a loose sense, the number of independent DEM values does not exceed the number of independent measurements (see, e.g., Craig & Brown 1986). There are, as well as these fundamental limitations of the DEM method, practical problems concerning the nature and magnitude of errors in the theoretical calculation of the intensities of emission lines. For example, Judge et al. (1995) showed that the $\xi(T_e)$ problem also has large sources of systematic error in excess of known errors in line intensities, which they suggested are due to the breakdown of the fundamental assumption of ionisation equilibrium made in formulating the problem, although radiative transfer effects could not be ruled out. In addition Judge et al. (1997) concluded that systematic errors in the atomic physics, and in the ionisation balance, make straightforward inversion for $\mu(n_e, T_e)$ very difficult or intractable.

The bivariate DEM function, $\mu(n_e, T_e)$, is the "holy grail" of solar UV spectroscopy but the difficulties clearly noted, and demonstrated, in Hubeny & Judge (1995) and Judge et al. (1997) mean that we are essentially limited, by the need for an element of uniqueness, to inversions for $\xi(T_e)$ and in the extreme for $\zeta(n_e)$ of Almleaky et al. (1989) and Brown et al. (1991)), or to $\mu(n_e, T_e)$ on a very coarse grid.

The 'mean value' or 'line ratio' approach on the other hand gives well defined results

which are appealing because they are simple to derive and they can remove, through careful choice of lines, large sources of uncertainty arising from errors in ionisation balance. However, they have the serious drawback that the results are not easy to interpret for inhomogeneous plasmas, different line ratios for example giving different mean densities even for lines peaking in the same temperature range because of their different detailed sampling of the temperature distribution (cf. Almleaky et al. 1989 and Brown et al. 1991). It is worth acknowledging that Brown et al. (1991) found that as the number of ratios used is increased the ratio estimates asymptotically approach their physcial values.

The exact relationship between the two approaches has never been explored in depth, although particular situations were discussed by Brown et al. (1991). Motivated by this, by the advent of new data from the CDS and SUMER instruments on the SOHO spacecraft, and by the desire to remove the large sources of systematic error that plague inversions of emission line data (e.g., Judge et al. 1995; Judge et al. 1997), we study the relationship between these two methods. We show below that there is a precise correspondence between DEM functions and a suitable complete set of mean spectroscopic densities and/or temperatures in situations where these can be defined.

After establishing this exact relationship between the line ratio and full DEM inversion techniques we pursue a method using the best features of both methods to improve the stability of inversions for $\xi(T_e)$, $\zeta(n_e)$ and $\mu(n_e, T_e)$. In Section 4.2 we demonstrate that such a 'hybrid' method alleviates the conditioning and stability effects introduced in Chapter 2 and in Judge et al. (1997). Stated simply, we propose a novel 'fusion' of these two basic techniques, the Ratio Inversion Technique (RIT), that can achieve numerically stable, and hence more reliable, source functions of the emitting region of the solar plasma. The RIT algorithm is discussed in Section 4.2.2 while results for several model DEM functions (for $\xi(T_e)$ and $\zeta(n_e)$) are discussed in Section 4.3. In Section 4.4,to complete our analysis, we apply the RIT to solar active region spectra obtained by the Solar EUV Rocket Telescope (SERTS) in 1989 (Thomas & Neupert 1994) to recover a form for $\xi(T_e)$. We compare the resulting form of $\xi(T_e)$ with those obtained independently by Brickhouse et al. (1995), Landi & Landini (1997) and Lanzafame et al. (1998).

4.1 Relation between line ratio and emission measure analyses

Before commencing with the derivation of the formal relationship between a set of spectroscopic mean values from emission line ratios and the emitting plasmas source function (the DEM functions discussed above) we must re-address, and complete, some of the concepts introduced in the previous chapter. In particular the details of optically thin line emission at or near coronal ionisation equilibrium in a highly non-LTE plasma. We begin by considering the total power P_i radiated by a particular spectral line labelled i. So, for an optically thin plasma occupying a volume V is

$$P_i = \int\int\int_V h\nu_i A_i n_{u(i)} dV \quad \text{erg s}^{-1} \tag{4.1}$$

where h is Planck's constant, ν_i is the frequency of the line, A_i (s^{-1}) is the Einstein A-coefficient, and $n_{u(i)}$ (cm^{-3}) is the population density of the upper level $u(i)$. Following standard practice, we define a line emission coefficient, $K_i(n_e(\mathbf{r}), T_e(\mathbf{r}))$, normalised to the electron density squared as

$$K_i(n_e(\mathbf{r}), T_e(\mathbf{r})) = \frac{h\nu_i}{4\pi}\frac{n_{u(i)}A_i}{n_e^2} \quad \text{erg cm}^3 \text{ sr}^{-1} \text{ s}^{-1}, \tag{4.2}$$

then equation (4.1) becomes

$$P_i = 4\pi \int_V K_i(n_e(\mathbf{r}), T_e(\mathbf{r}))\, n_e^2(\mathbf{r})\, d^3\mathbf{r} \quad \text{erg s}^{-1}. \tag{4.3}$$

We remind the reader that to write the equation in this form we have made several implicit assumptions, these are assumptions are stated in Chapter 2. We have observed that $K_i(n_e(\mathbf{r}), T_e(\mathbf{r}))$ is almost independent of density n_e for collisionally excited permitted transitions decaying to the ground state of a given ion.

Equation (4.3), with full dependence on n_e and T_e, can be formulated in terms of a function of electron density and temperature. This function was identified above as the bivariate DEM function of n_e and T_e, namely $\mu(n_e, T_e)$. If we follow the procedure used to formulate equation (2.68) for the line intensity I_i and not the total radiated power P_i, we have $I_i = P_i/(4\pi S)$, where S is the area of the projected volume V and

$$I_i = \int_{T_e}\int_{n_e} K_i(n_e, T_e)\mu(n_e, T_e)\, dn_e dT_e \quad \text{erg cm}^{-2} \text{ sr}^{-1} \text{ s}^{-1}. \tag{4.4}$$

Again, we define the differential emission measure in n_e, $\zeta(n_e)$, as the reciprocal density-gradient-weighted mean square electron density and, correspondingly the differential emission

measure in T_e, $\xi(T_e)$ as the reciprocal temperature-gradient-weighted mean square electron density, obtained from equation (2.67), as follows:

$$\zeta(n_e) = \int_{T_e} \mu(n_e, T_e) dT_e \quad \mathrm{cm}^{-2} \tag{4.5}$$

$$\xi(T_e) = \int_{n_e} \mu(n_e, T_e) dn_e \quad \mathrm{cm}^{-5}\ \mathrm{K}^{-1} \tag{4.6}$$

As discussed previously, in the context of inverse methodology, these moments are the best way to interpret the raw observed data to help determine the energy balance (Jordan et al. 1987; Griffiths & Jordan 1998) or determining whether the data are compatible with an atmosphere at constant pressure (see Craig & Brown 1976; Judge et al. 1997). Formulation of the relationship between these functions and the 'mean' observed quantities is therefore of benefit to the solar physics community.

4.1.1 Relationship between $\xi(T_e)$ and $\langle T_e \rangle$

Consider an optically thin emission line labelled i for which $K_i(n_e, T_e)$ is a weak function of density, such as a resonance line. $K_i(n_e, T_e)$ can then be replaced by $K_i(T_e)$ (as discussed in the previous chapter). So, we have the spectral line intensity (in the appropriate units)

$$I_i = \int_{T_e} K_i(T_e)\, \xi(T_e) dT_e\,. \tag{4.7}$$

For two such lines i, j, the ratio of the two line intensities is

$$R_{ij} = \frac{I_i}{I_j} = \frac{\int_{T_e} K_i(T_e)\, \xi(T_e)\, dT_e}{\int_{T_e} K_j(T_e)\, \xi(T_e)\, dT_e}\,, \tag{4.8}$$

and if the emission coefficients are different, then the ratio depends on T_e. If the plasma is homogeneous in temperature, i.e. isothermal, we can express the $\xi(T_e)$ as a Dirac delta function $\xi(T_e) = \xi_0\, \delta(T_e - \langle T_e \rangle)$ such that, on substituting this expression into to equation (4.8) and integrating over the whole temperature domain, we have

$$R_{ij} = \frac{\xi_0\, K_i(\langle T_e \rangle)}{\xi_0\, K_j(\langle T_e \rangle)} \tag{4.9}$$

which is trivially reduced (on dividing throughout by ξ_0) to express R_{ij} in terms of the 'mean' spectroscopic temperature, $\langle T_e \rangle_{ij}$. So again, for our particular line pair (i, j) we have

$$R_{ij} = \frac{K_i(\langle T_e \rangle)}{K_j(\langle T_e \rangle)} = S_{ij}(\langle T_e \rangle_{ij}) \tag{4.10}$$

where $S_{ij}(T_e) = \frac{K_i(T_e)}{K_j(T_e)}$ is assumed to be a monotonic, bijective (invertible) function which has a unique inverse on the temperature domain considered when we restrict our study to

resonance lines, i.e. different T_e and *no* dependence on n_e, only (see, e.g., figure 2.9). For these conditions the relationship is almost always satisfied. Therefore, on inspection, the relation between $\langle T_e \rangle_{ij}$ and the observed line ratios R_{ij} is given by

$$\langle T_e \rangle_{ij} = S_{ij}^{-1}(R_{ij})\,. \tag{4.11}$$

To formulate an expression for $\xi(T_e)$ in terms of the 'mean' spectroscopic temperatures we must return to equation (4.7). On dividing through equation (4.7) by any other line intensity $I^{obs}_{j(i)}$, $(i \neq j)$, known to depend differently on T_e from line i (hence the notation $j(i)$), we obtain

$$R^*_{i,j(i)} = \frac{I_i}{I^{obs}_{j(i)}} = \int_{T_e} K'_i(T_e)\,\xi(T_e)\,dT_e \tag{4.12}$$

with $K'_i(T_e) = \frac{K_i(T_e)}{I^{obs}_{j(i)}}$. This expression thus gives the ratio of the *theoretical* intensity for line i to the *observed* intensity of line $j(i)$. At this stage I_i, and hence $R^*_{i,j(i)}$, are not known quantities. If we set $R^*_{i,j(i)} = R_{i,j}$, the observed line ratio, then equation (4.12) becomes an integral equation with known LHS, and known kernel $K'_i(T_e)$, in which $\xi(T_e)$ is the quantity to be determined. Consider forming n ratios of the intensities of a set of emission lines to form a vector $\underline{R}$:

$$\underline{R} = (R_{1,j(1)}, R_{2,j(2)}, ..., R_{n,j(n)}) \tag{4.13}$$

If we discretise equation (4.12) with respect to T_e, then the equation becomes a matrix equation of the form:

$$\underline{R} = \underline{\underline{K'}}\ \underline{\xi}. \tag{4.14}$$

The rows of $\underline{\underline{K'}}$ are simply rows of kernels of equation (4.7) divided by observed line intensities. This has the (poorly conditioned, see Craig & Brown 1986) analytical solution:

$$\underline{\xi} = \underline{\underline{K'}}^{-1}\underline{R}. \tag{4.15}$$

This equation for n ratios permits $\underline{\xi}$ to be determined at up to n discrete temperatures. The above illustrates that the equations for line ratios can be simply re-written in a standard form, which can thus be used in numerical algorithms and will be discussed below. But we have not yet written the formal equivalence between the $\xi(T_e)$ functions and a set of line ratios, and their corresponding mean derived temperatures. From the above, this is clearly just

$$\underline{\xi} = \underline{\underline{K'}}^{-1}\underline{\{S_{ij}(\langle T_e \rangle_{ij})\}}, \tag{4.16}$$

where $\underline{\{S_{ij}(\langle T_e \rangle_{ij})\}}$ denotes the array of line ratios indexed by i. This expression relates the DEM to the set of spectroscopically derived temperatures through the inverse of the matrix $\underline{\underline{K'}}$.

4.1.2 Relationship between $\zeta(n_e)$ and $\langle n_e \rangle$

Due to the inhomogeneous nature of the solar atmosphere it is clear that the constituent plasma has no unique n_e. We can, never the less, define a spectroscopic 'mean' electron density for the ratio of lines displaying some degree of density sensitivity. As stated previously the ideal ratio being that of a resonance (essentially no n_e functional dependence) and an intersystem (with functional behaviour with n_e categorised earlier) line of the same ionisation stage of a particular atom.

To obtain such a 'mean' estimate of n_e we must consider the optically thin plasma to be isothermal, with $T_e = T_0$, meaning that $K_i(n_e, T_e = T_0)$ reduces to $K_i(n_e)$. Then, the total emitted line intensity of a line labelled i is given by

$$I_i = \int_{n_e} K_i(n_e)\, \zeta(n_e)\, dn_e \,. \tag{4.17}$$

Then for a density sensitive line pair (i, j) we see that the ratio R_{ij} is given by (cf. equation (4.8))

$$R_{ij} = \frac{I_i}{I_j} = \frac{\int_{n_e} K_i(n_e)\, \zeta(T_e)\, dn_e}{\int_{n_e} K_j(n_e)\, \zeta(T_e)\, dn_e} \,. \tag{4.18}$$

As above, we now seek the 'mean' electron density $\langle n_e \rangle$ of a homogeneous plasma that yields the same line ratio as the observed inhomogeneous one. This is performed by defining $\zeta(n_e) = \zeta_0\, \delta(n_e - \langle n_e \rangle)$ such that, on substituting into equation (4.18) and dividing throughout by ζ_0 we have

$$R_{ij} = \frac{K_i(\langle n_e \rangle)}{K_j(\langle n_e \rangle)} \,. \tag{4.19}$$

By direct analogy to the steps producing equations (4.8) through (4.16) we can construct a relationship for the discretised differential emission measure in n_e, $\underline{\zeta}$, in terms of a set of 'mean' spectroscopic densities $\underline{\langle n_e \rangle_{ij}}$, and the operator $G_{ij}(n_e) = \frac{K_i(n_e)}{K_j(n_e)}$. For purposes of writing expressions formally equivalent to those above, this operator must now be *assumed* to be unique (monotonic, bijective). Thus,

$$\underline{\zeta} = \underline{\underline{K'}}^{-1}\, \underline{\{G_{ij}^{-1}(\langle n_e \rangle_{ij})\}}, \tag{4.20}$$

where $\underline{\underline{K'}}^{-1}$ is to be understood as the equivalent (but clearly not identical) matrix to that in equation (4.16). While this expression assumes that the inverse operator $G_{ij}^{-1}(\langle n_e \rangle_{ij})$ has

a unique solution, notice that a numerical solution for $\underline{\zeta}$, analogous to equation (4.15), makes no such assumption. In fact, it removes ambiguities that can arise from the non-unique inverse operator $G_{ij}^{-1}(\langle n_e \rangle_{ij})$ for certain line ratios in important ions. This is because, in a numerical implementation, this operation is not performed. The vector element is instead set to the observed ratio $R_{i,j(i)}$. **An example of non-unique inverse operators occurs for certain ratios of intersystem lines in the boron isoelectronic sequence (see, e.g., Brage et al. 1996, Fig. 2).**

4.1.3 Relationship between $\mu(n_e, T_e)$ and $\langle n_e \rangle$, $\langle T_e \rangle$ pairs

In the general case we wish to obtain information about the form of the bivariate differential emission measure, $\mu(n_e, T_e)$ from a set of 'mean' spectroscopic densities, $\langle n_e \rangle$, and temperatures, $\langle T_e \rangle$, discussed above. These 'mean' values are usually derived individually, as described earlier, by looking at line pairs that are mostly sensitive to T_e, or n_e, but not both.

Following the method of the previous sections, we seek mean parameters $\langle n_e \rangle$ and $\langle T_e \rangle$ of the homogeneous plasma that will yield the same line ratio as the observed inhomogeneous plasma. Some care must be taken here, as can be seen by, following earlier sections, assuming that the bivariate DEM function can be approximated as separable by $\mu(n_e, T_e) = \mu_0 \delta(T_e - \langle T_e \rangle)\delta(n_e - \langle n_e \rangle)$. Using equation (4.4) to form the line ratio of two lines with labels i and j, $(i \neq j)$:

$$R_{ij} = \frac{I_i}{I_j} = \frac{\int_{T_e}\int_{n_e} K_i(n_e, T_e)\mu(n_e, T_e)\, dn_e dT_e}{\int_{T_e}\int_{n_e} K_j(n_e, T_e)\mu(n_e, T_e)\, dn_e dT_e} \tag{4.21}$$

On substitution of $\mu(n_e, T_e)$ given above into equation (4.21) and performing the double integral we obtain

$$R_{ij} = \frac{K_i(\langle n_e \rangle, \langle T_e \rangle)}{K_j(\langle n_e \rangle, \langle T_e \rangle)} = M_{ij}(\langle n_e \rangle, \langle T_e \rangle) \tag{4.22}$$

To try to determine $\langle n_e \rangle$ and $\langle T_e \rangle$ does not make sense, since there is just one equation, but two unknowns, $\langle n_e \rangle$ and $\langle T_e \rangle$. Thus it is clear that another equation is needed. One possible solution is to assume that $\langle T_e \rangle = T_{ij}^0$ where T_{ij}^0 is the coronal ionisation equilibrium temperature for the particular ion(s) under study. This is in fact a common assumption made for solar corona lines (e.g., Mason 1991). If this assumption (or something else) is made, then for a set of emission lines of temperature and density sensitivity, we see that the pair $(\langle n_e \rangle, \langle T_e \rangle)$ can be determined provided there exists an inverse function M_{ij}^{-1}, *i.e.*

$$(\langle n_e \rangle_{ij}, T_{ij}^0) = M_{ij}^{-1}(R_{ij}) \tag{4.23}$$

Repeating the steps taken to formulate equation (4.12) we divide through equation (4.4) by another line intensity, $I_{j(i)}$, again displaying the required functional (either density sensitive or temperature sensitive) behaviour to produce:

$$R_{i,j(i)} = \frac{I_i}{I_{j(i)}} = \int_{T_e}\int_{n_e} K_i'(n_e, T_e)\mu(n_e, T_e)\, dn_e dT_e \tag{4.24}$$

discretising this with respect to n_e and T_e we have the following

$$R_{i,j(i)} = \sum_{l=1}^{m}\sum_{q=1}^{p} \mu(n_q, T_l)\, K_i'(n_q, T_l)\, \Delta n_e\, \Delta T_e \tag{4.25}$$

Performing an operation described in Hubeny & Judge (1995) we re-index from $l = 1, \ldots, m$ and $q = 1, \ldots, p$ to $\kappa = 1, \ldots, mp$ so that equation (4.25) may be recast in a standard matrix form, where the Δn_e, ΔT_e terms are combined to form a measure of the redimensioned space, namely $\Delta(N_e \bigotimes T_e)$ and absorbed into the redimensioned form of $K'(n_e, T_e)$. Therefore equation (4.25) becomes:

$$R_{i,j(i)} = \sum_{\kappa=1}^{mn} U_\kappa\, K_{i\kappa}' \tag{4.26}$$

Where $\underline{\mathrm{U}}$ is the 1 dimensional transform of the 2 dimensional function $\underline{\underline{\mu}}$. This has an analytical solution of the form (cf. equations (4.16) and (4.20))

$$\underline{U} = \underline{\underline{K'}}^{-1}\underline{\{M_{ij}^{-1}(\langle n_e\rangle_{ij}, T_{ij}^0)\}}. \tag{4.27}$$

Again $\underline{\underline{K'}}^{-1}$ is equivalent, but not equal to that of equation (4.16). Also, the comments above on the uniqueness of inverse operators in the $\underline{\zeta}$ problem apply equally to the bivariate problem.

4.2 Ratio inversion solutions for DEM functions

The relationships discussed in the previous sections have alluded to a 'clean' relationship between line ratios and distributions of the fundamental plasma quantities (i.e. the DEM functions). In the following discussion we will show that these relationships can be taken one step further using a new, hybrid algorithm to obtain the discretised DEM functions using a line ratio-like inversion method. Such an approach has several advantages over either of the two methods previously discussed. The principal disadvantages of both approaches are, as discussed above:

1. Using the individual line ratio approach on its own is not enough to obtain meaningful reliable distributions of the plasma diagnostic quantities.

2. The full solution of the ill-posed inverse problem to obtain the DEM functions is very much influenced by theoretical uncertainties in the atomic factors used to formulate the problem and not only observational errors.

It is important to stress that the *only* standard inversion carried out that considered such theoretical uncertainties was Judge et al. (1997). Also, the systematic nature of these uncertainties require that a method like the RIT is needed.

The difficulty with 'fusing' these two concepts to produce a hybrid algorithm is of a purely conceptual nature. Standard practice when solving an inverse problem is, as discussed in Chapter 2, a matter of constructing a linear matrix equation. Not only this but we are required to use some form of regularising mechanism to constrain the smoothness of the recovered solution. Mathematically speaking, we are attempting to solve the ratio of two integral equations for a univariate DEM function, $f(s_e)$, of the observed diagnostic quantity s_e (n_e or T_e), each given by (cf. equations (4.7) and (4.17))

$$I_i = \int_{s_e} K_i(s_e) f(s_e) ds_e \; . \tag{4.28}$$

Recalling that this equation can be expressed as a linear matrix equation $\mathbf{g} = K\,\mathbf{f}$ and that the errors δK in the line emissivities can be transported, via the errors in line intensities δg, to fractional errors and numerical instabilities in the recovered solution (assuming that $\mathbf{f} = K^{-1}\mathbf{g}$ exists) of the form, cf. equation (2.28), for an arbitrary norm

$$\frac{\|\delta\mathbf{f}\|}{\|\mathbf{f}\|} \leq \left(\frac{C_K}{1 - C'_K}\right) \frac{\|\delta\mathbf{g}\|}{\|\mathbf{g}\|} + \left(\frac{C'_K}{1 - C'_K}\right) \tag{4.29}$$

where C_K and C'_K are the condition number and adjusted condition number of matrix K as defined in Chapter 2. So, for a set of N line ratios $\{\ R_{ij}\ \}$ we have for line pairs i, j (and $i \neq j$) with respective integrated line intensities I_i and I_j

$$R_{ij} = \frac{I_i}{I_j} = \frac{\int_{s_e} K_i(s_e) f(s_e)\, ds_e}{\int_{s_e} K_j(s_e) f(s_e)\, ds_e} \; . \tag{4.30}$$

From this relationship we can envision why the line ratio technique has proven so popular as the main diagnostic technique in space borne ultraviolet spectroscopy; the ratio of like atomic terms negates systematic errors in those terms. This is one of the points this work will exploit. Similarly, we use the rigour[1] of obtaining the DEM functions as eloquently and pointedly stated by Pottasch (1964) and later by Craig & Brown (1976).

[1] Of course, this is rigour in the mathematical sense; the DEM functions are the *only* true diagnostic of the emitting solar plasma from such an inverse formalism.

Recently, Fludra & Schmelz (1995) employed a line-ratio approach, loosely comparable to the RIT, to infer coronal atomic abundances of the flaring coronal plasma. Their discussion focused on the analysis of Soft X-ray (10 - 100 keV) lines obtained by the Solar Maximum Mission (SMM) Flat Crystal Spectrometer (FCS; Acton et al. 1980) and produced, as a by--product, DEM functions $\xi(T_e)$ for the high temperature ($6 \leq \log_{10} T_e \leq 8$) flaring plasma. The analysis Fludra & Schmelz (1995) presented however, did not make any effort to compensate for the potentially damaging theoretical atomic uncertainties discussed above. Even though the community was well aware of the difficulties of constructing reliable atomic transition models it was never properly addressed in the literature until 1997 with the work of Judge et al. (1997).

In an ideal world, one where the solution to equation (4.30) is a smooth positive definite function of s_e, we would seek the least squares solution (cf. Section 2.1) of

$$X^2(R_{obs}, R_{calc}) = \sum_{l=1}^{N} \left(\frac{\left(R_l^{obs} - R_l^{calc} \right)^2}{\sigma_{l_{th}}^2 + \sigma_{l_{obs}}^2} \right) \tag{4.31}$$

where l is the label of a particular line pair, $\{ R_{obs} \}$ is the set of observed optically thin line ratios with errors $\sigma_{l_{obs}}$, *theoretical* estimates of the errors in the relevant atomic parameters (in $K_i(s_e)$ and $K_j(s_e)$) given by $\sigma_{l_{th}}$ (discussed below) and the set of $\{ R_{calc} \}$ are calculated using equation (4.30). However, as is the case with all ill-posed inverse problems, we must seek a regularised solution for $f(s_e)$ which minimises (adopting X, above, to be a *form* of the statistical χ^2 measure between R_{obs} and R_{calc})

$$\chi^2 = X^2(R_{obs}, R_{calc}) + \lambda \, \Phi(f(s_e)) \tag{4.32}$$

where λ and $\Phi(f(s_e))$ are the smoothing parameter and smoothing functional respectively. Clearly, where equation (4.32) is non-linear in $f(s_e)$, from the $X^2(R_{obs}, R_{calc})$ term, the linear case solving for a set of line intensities requires that we solve equation (2.46). As we have previously discussed the choice of $\Phi(f(s_e))$ will reflect the nature of the solution space, for example considering $f(s_e)$ to be smooth to the n^{th} polynomial order such that

$$\Phi(f(s_e)) = \int_{s_e} \left| \frac{d^n f(s_e)}{ds_e^n} \right|^2 ds_e \tag{4.33}$$

where $f(s_e)$ will clearly be a discretised function and we will be required to calculate equation (4.33) as a forward finite-difference estimate of the actual integral.

The form of equations (4.31) and (4.32) mean that we *cannot* use standard Tichonov regularisation or SVD methods, but that we *must* adopt a new non-linear approach. To

this end we have chosen a Genetic Algorithm (GA) because of its numerical robustness (Goldberg 1989) and the ease with which non-linear calculations like equation (4.32) can be encoded (Charbonneau 1995). The terminology and basic mechanical principles of GAs were discussed in the previous chapter. For the calculations presented here we will specify the s_e mesh over which the integrals are discretised and use different smoothing functionals (over a wide range values for λ) to analyse the numerical stability of the solutions obtained. Indeed we show that, for a series of test DEM 'source' functions, the results are conclusive that this method, the Ratio Inversion Technique (RIT), is not influenced greatly by large systematic errors in the atomic rate coefficients that could make the results of standard intensity inversions highly ambiguous. **That is, the RIT is insensitive to errors that are likely to dominate standard inversion procedures and therefore provides a new means of obtaining less ambiguous results about the emitting optically thin region of the solar atmosphere under examination.**

4.2.1 Calculation of kernel errors

To calculate meaningful values of $\sigma_{l_{th}}$ (for each line pair l) we have performed a Monte Carlo simulation to get a distribution of twenty perturbed line emissivities for each transition. Perturbed, in the sense that their component atomic terms (rates and coefficients) are randomly perturbed about their "accepted" values. The amounts by which these coefficients and rates are perturbed are relevant to figures put forward in the literature, specifically in Judge et al. (1995) and Judge et al. (1997). Recalling from Section 2.2 that we can express the emissivity of the optically thin emission line i (in the simplest sense) as

$$K_i(s_e) \approx \kappa \cdot \mathcal{X}_i(s_e) \cdot \mathcal{Y}_i(s_e) \tag{4.34}$$

where κ is a constant, $\mathcal{X}_i(s_e) = \frac{n_{ion}}{n_{el}}$ is the conglomerate of the bound-free (**b-f**) terms and $\mathcal{Y}_i(s_e) = \frac{n_{u(i)}}{n_{ion}}$ of the bound-bound (**b-b**) terms of the transition as functions of s_e respectively[2]. So, if these quantities have associated errors $\delta\mathcal{X}_i$ and $\delta\mathcal{Y}_i$ then the fractional error in the line emissivity can be expressed as

$$\left(\frac{\delta K_i}{K_i}\right)^2 = \left(\frac{\delta \mathcal{X}_i}{\mathcal{X}_i}\right)^2 + \left(\frac{\delta \mathcal{Y}_i}{\mathcal{Y}_i}\right)^2 \ . \tag{4.35}$$

The calculations presented in this chapter have associated standard (1σ) deviations in the fractional errors of the order (cf. Judge et al. 1997) :

[2]Of course these terms have been defined previously. The definition of $\mathcal{X}_i$ remains unchanged, but $\mathcal{Y}_i$ was defined as the elemental abundance relative to hydrogen.

- For the bound-bound processes we adopt a value of 3%. This of course ensures, by definition, that 32% of the random realisations will have errors in excess of 3%.

- We have chosen to use logarithmic (base 10; log-normal distributed) deviations of ± 0.1 about the mean value for bound-free processes. This value is clearly an estimate because the amplitude of errors in such (**b-f**) processes are not well known, P. G. Judge - Private Communication.

These values reflect possible *lower* magnitude limits on the **b-b** and **b-f** terms. So, the effects on line emissivity $K_i(s_e)$ are conservatively estimated to lie between 10% and 125%. Of course, there are other possible atomic and external mechanisms that can further increase these estimates but discussion of these is left until Chapter 6.

The actual process of perturbing the atomic rates/coefficients is carried out by routines of the HAO-Diaper atomic calculation package. To obtain actual estimates of $\sigma_{l_{th}}$ we have to obtain a distribution of line emissivities for each line, each with different random realisations of the constituent atomic factors. We obtain twenty such realisations for each line and use the following recipe to construct values of $\sigma_{l_{th}}$ for the line pair $l = (i, j)$.

1. Calculate the integrated line intensities for *each* line and each perturbed line emissivity; yielding a distribution of $Q = 20$ line intensities. It should be noted that we use a constant 'flat' $f_0(s_e)$ to calculate these intensities but such an approximation is not taken lightly and is made primarily to have a simple and uniform error estimate for every line no matter at what temperature it is formed at. So, returning to the problem in hand we have calculated 20 randomly perturbed values I_i' (cf. equation (4.28))

$$I_i' = \int_{s_e} K_i(s_e) f_0(s_e) ds_e \; . \tag{4.36}$$

2. Repeat the previous calculation for every possible line until distributions of line intensities $I_i = \{ \; I_i^1, \ldots, I_i^Q \; \}$ are formed.

3. Use the distributions of step 2 to form distributions for the various emission line pairs $R_l = \{ \; R_l^1, \ldots, R_l^Q \; \}$. Note that the individual values of R_l^j $(1 \leq j \leq Q)$ are calculated with the denominator and numerator line intensities are taken from the same model, model j.

4. Given now that we have a random distributions for the same 'flat' $f(s_e)$ function it is reasonable assume that the standard deviations (1σ) of the distributions approximate the values of $\sigma_{l_{th}}$ well.

4.2.2 Specifics of the Ratio Inversion Technique (RIT)

As noted above we are making use of the adaptability of a Genetic Algorithm (GA) to perform this non-linear inversion. The GA approach allows a very high degree of control to be placed in the hands of the user (i.e. us) and a GA effectively allows us to specify the number of generations (10,000; significantly more than the examples presented in Chapter 3) over which the solution will evolve over; the final solution being that which best optimises equation (4.32). Also, the GA method we use implements a genetic precedence operator known as *elitism*, its function being that the solution best satisfying equation (4.32) (at the end of each generation) is retained for in the population of possible solutions for *breeding* the next and preceding generations.

Each individual in the population, composed of 100 individuals, is made up of $M = 30$ 'parameters' with the i^{th} parameter evaluating the DEM function at the i^{th} point in s_e space, i.e. $f(s_{e_i})$. RIT does not couple these parameters (there is no interpolation between them) and choice of $M = 30$ as the number of discretisation points is entirely arbitrary. This number can be increased but care must be exercised because, as M increases the line emissivities get 'closer' to the continuous integral operators[3] they represent and increase the possibility of numerical instability. The choice of N, the number of line ratio pairs used in the analysis is arbitrary, but we note that significant increase in N above 30 say, may also produce an increase in numerical instability of the recovered solution. This is particularly true if using an increased number of ratio pairs from one particular ionic stage since then the 'linear dependence' of the operator to be inverted is increased considerably. This was discussed in Chapter 2 and shall be met again, in greater depth, in Chapter 5.

The action of the RIT is best described as the following :

1. Generate 100 random solutions as an initial population, calculate the resulting χ^2 of equation (4.32) for each individual.

2. Choose a subset according to their values of χ^2, and breed them to produce a new population.

3. Calculate the value of χ^2 for each individual in the new population.

4. Replace the old population with the new one.

[3]Not only do they 'approach' the actual form of the integral operators, a patently poor property, but they reduce the effectiveness of the genetic operators; this is discussed as earlier, see e.g, Chapter 3.

5. Check that the number of generations has reached its maximum value; if not return to step 2.

4.3 Results

In this section we highlight the operation of the RIT algorithm on several test DEM functions and compare the results with standard (linear) inversions. Section 4.3.1 discusses the analysis of the emission measure differential in temperature, $\xi(T_e)$, to all intents and purposes the singly most important diagnostic quantity in terms of inverse modelling the solar atmosphere. Section 4.3.2 discusses application of the RIT to emission measure differential in electron density, $\zeta(n_e)$ in a similar vain to that of $\xi(T_e)$ especially in that the performance of both 'flavours' of ratio inversion will be compared to those obtained using standard intensity inversions[4].

It is important at this stage to note that the resultant recovered (discretised) function $f(s_e)$ from optimising equation (4.32) (through equation (4.31)) does *not* allow us to fix the amplitude of $f(s_e)$. Simply because, when returned it will always be a multiple $\mathcal{C}$ of its true value since

$$R_{calc} = \frac{\int_{s_e} K_i(s_e)\,(\mathcal{C}f(s_e))\;ds_e}{\int_{s_e} K_j(s_e)\,(\mathcal{C}f(s_e))\;ds_e} \tag{4.37}$$

will *always* hold. So, to resolve this problem we must use the recovered solution $f(s_e)$ to re-calculate the M $(< 2N)$ line intensities, I_{calc}. These "new" intensities, when compared to the observed intensities, yield the scaling factor $\mathcal{S}(\approx \mathcal{C})$ given by

$$\mathcal{S} \;=\; \frac{1}{M}\;\left(\sum_{j=1}^{M}\;\frac{I^j_{obs}}{I^j_{calc}}\right)\;. \tag{4.38}$$

In practice, we need only fix a single line intensity in the calculation to fix the absolute magnitude of $f(s_e)$ but, in the presence of noise, this leads to an element of *bias* in the function scaling. This possible source of bias is because any particular line intensity is only 'sensitive' over a short span of the whole s_e domain - from the s_e dependence of the line emissivity - the concept of 'emissivity coverage' is discussed in greater detail in Chapter 5. Thus, equation (4.38) yields an unbiased scaling factor for $f(s_e)$ by effectively averaging out the scaling over the entire s_e domain.

As is clear from the discussion of Section 2.1, knowledge of an optimal value λ for the inversion — in other words one which best reproduces the data (the "X^2" term) — but with

[4]The intensity inversions described here will be performed using a Tichonov regularisation routine with the same polynomial order of smoothing as used by the RIT to ensure a fair comparison.

a sufficiently smooth function $f(s_e)$ is important. For the standard Tichonov inversions it is simple to estimate a value for λ

$$\lambda \sim \frac{tr(K^T\,K)}{tr(H)} \tag{4.39}$$

where K and H are the kernel and smoothing matrices respectively. How to obtain such an optimal value of λ is not so clear when using the RIT because of the non-linearity of the operator and as in the application of the scaling function above we must address this difficulty *a posteriori*. That is, we must obtain solutions that minimise equation (4.32) over a wide range of λ. This is performed by extending the estimation method of linear inversions (cf. figure 2.3) and considering a reformulation of equation (4.32)

$$\chi^2 = X^2 + \lambda D^2 \tag{4.40}$$

where D is the evaluation of the smoothing operator. The simplest way to consider this is graphically, i.e. we plot λ versus D^2 versus χ^2 - the vector $\mathbf{v} = (\lambda, D^2, \chi^2)$. As an extension to the standard linear inversion case the best solution, that not only satisfying equation (4.32) but having a smooth functional form is given by

$$\lambda_{opt} = \min_{\lambda}\{\sqrt{\lambda^2 + \chi^2 + D^2}\} = \min_{\lambda} \|\mathbf{v}\|_2 \tag{4.41}$$

where the Euclidean norm $\|\cdot\|_2$ is as previously defined.

It is important to stress at this point that, for the test model $f(s_e)$ functions considered, we are taking a 'forward-backward' approach. That is, for a specific model $f(s_e)$ function, we perform the following steps

Forward - For the series of lines from which we will eventually construct line ratios we must calculate, for a specific model $f(s_e)$, integrated line intensities (equation (4.28)). We then randomly perturb these line intensities wiht a 1σ error of 15%. These intensities are then used to construct the line ratios used in the RIT, in this case R_l^{obs} for line pair $l = (i, j)$.

Backward - Taking these values for R_l^{obs}, their errors $\sigma_{l_{obs}}$ and the values of $\sigma_{l_{th}}$, calculated as described in Section 4.2.1, we then seek to optimise equation (4.32) for specific values of λ and smoothing functional $\Phi(f(s_e))$.

As a fair test of the RIT we perform the backward 'leg' twice, once using the emissivities used to calculate the intensities and the other using perturbed line emissivities. The point being that, if the errors are truly systematic then the ratio analysis should alleviate their numerical effect on the recovered solution. These are hereafter referred to as *standard* and

perturbed inversions. Note that the perturbed inversions are all carried out with the same set of perturbed emissivities which is randomly chosen from the set of 20 discussed above.

Using these we will see how well the RIT 'filters out' systematic errors in the line emissivities which cause the catastrophic instabilities for standard intensity inversions found by Judge et al. (1997). The following subsections detail the results of these tests for $\xi(T_e)$ (Section 4.3.1) and $\zeta(n_e)$ inversions (Section 4.3.2) and, for 'completeness', Section 4.3.3 demonstrates the usefulness of implementing a different, more generalised smoothing (or regularisation) functional.

4.3.1 RIT test results for $\xi(T_e)$

In this section we test the properties of the Ratio Inversion Technique (RIT) against those of a 'standard' inversion method. That is, we will exploit the error filtering capabilities of the line-ratio technique of obtaining plasma diagnostic quantities and numerical instability of a GA optimisation approach in the face of large systematic errors in the line emissivities like those discussed in Section 2.2.

The model $\xi(T_e)$ functions we consider here are contrived to encompass the various classes of $\xi(T_e)$ function likely to occur in the solar atmosphere and not as having any specific physical plasma interpretation. We study two such functions here: model 1 is strictly continuous over the temperature domain ($4.5 \leq \log_{10} T_e \leq 6.5$) and is parabolic in form whereas model 2 is a 'Top-Hat' function with discontinuities in T_e. The two test models are shown in figure 4.1, recall that these functions are discretised over 30 temperature points.

Table 4.1 identifies the line ratio pairs l used in these calculations along with their wavelengths (λ Å) and their measure of the theoretical uncertainty in the line ratio ϵ_l ($\sigma_{l_{th}}$ as a fraction of $R_{l_{th}}$ for a flat model $\xi(T_e)$ function.) It is clear that, as anticipated, the line ratio pairs with each line belonging to a common ionisation stage of the atom having considerably lower values, in general, than others being typically in the range of $\epsilon_l \approx 2 - 10\%$. The 'Uncorrelated' line ratio pairs have typical values upward of 20%. Note, the unusually large error in the ratios of lines within the Lithium like (Li-like, to use the notation of Chapter 2) ion C IV. Figure 4.2 illustrates the discrepancy between two different atomic models through the ionisation balance of multiply ionised carbon (C II - C IV) as a function of T_e.

As stated in the previous section the solutions are evolved over a fixed number of generations (10,000) over a wide range of values for λ and for different orders n of smoothing or regularisation ($n = 1$ or 2 for these simple test cases). These tests allow analysis of the RIT in

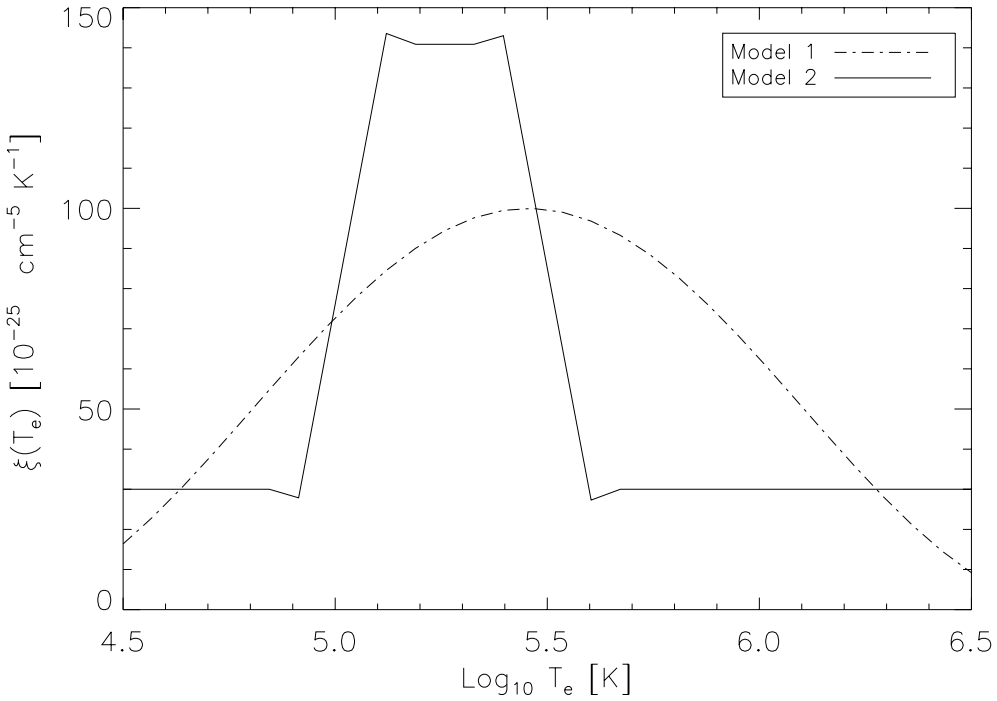

Figure 4.1: Plot of the two test model forms of $\xi(T_e)$. Model 1 (dashed line) is a continuous function whereas model 2 (solid line) has two discontinuities in T_e. These two model functions display all the major characteristics that we may see in real inferred $\xi(T_e)$ functions of the solar atmosphere.

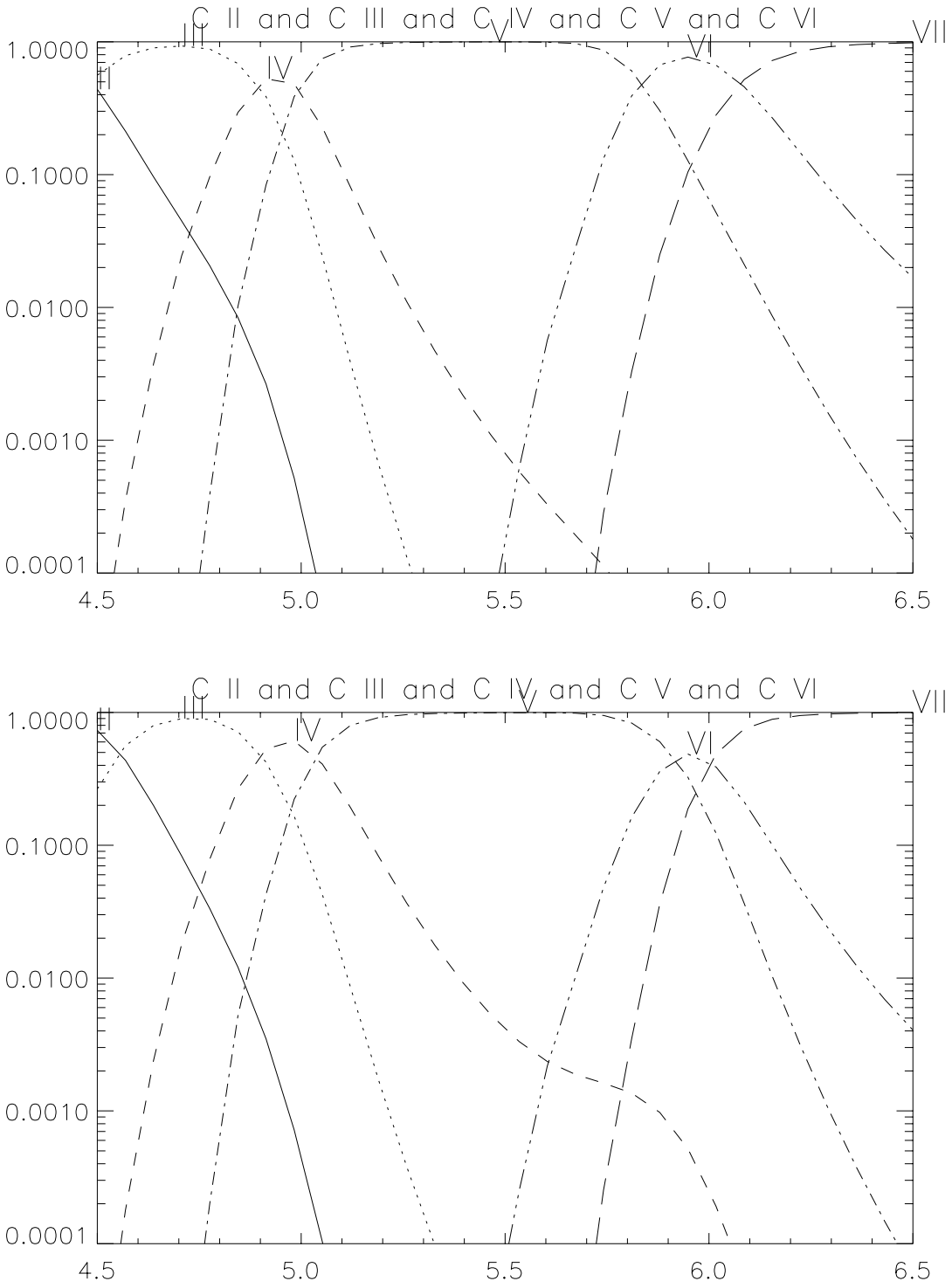

Figure 4.2: The ionisation fractions of two different atomic models for Carbon ions C II through C VI, (top) a standard unperturbed model, and (bottom) a model where rate coefficients have been subject to random perturbations. The effect of the perturbation is most clearly seen in 'Li- like' C IV ion. The horizontal scale is $\log_{10} T_e$.

Table 4.1: Details of the line pairs used in the RIT runs on $\xi(T_e)$ presented in this chapter. For each ratio pair $l = (i, j)$ of R_{ij} the numerator, i, (N) and denominator, j, (D) lines are indicated, along with the ionic stage to which they belong and their wavelength (λ Å). Also quoted is the measure of uncertainty ϵ_l (i.e. $\sigma_{l_{th}}$ as a fraction of the theoretical line ratio $R_{l_{th}}$ for a flat model DEM) from the distribution of 20 perturbed line emissivities. Ratio pairs 1 through 22 are known here as 'Correlated' ratios since they have errors in **b-b** rates only whereas pairs 23 through 30 are 'Uncorrelated' and include errors in the **b-f** rates also.

#	Ion_N	λ_N	Ion_D	λ_D	ϵ_l	#	Ion_N	λ_N	Ion_D	λ_D	ϵ_l
1	C IV	1548.18	C IV	312.420	0.1482	2	C III	977.020	C III	1175.26	0.0472
3	Mg IX	706.060	Mg IX	368.070	0.0397	4	Mg IX	706.060	Mg IX	445.980	0.0273
5	Ne VII	895.175	Ne VII	465.220	0.0550	6	Ne VII	895.175	Ne VII	562.993	0.0194
7	Ne VI	562.711	Ne VI	999.630	0.0525	8	Ne VI	562.711	Ne VI	454.072	0.0566
9	Si III	1206.49	Si III	1301.14	0.0671	10	N III	991.502	N III	772.385	0.0299
11	O VI	1031.91	O VI	150.089	0.0966	12	O V	1218.34	O V	629.732	0.0405
13	O V	1218.34	O V	761.128	0.0352	14	O IV	790.112	O IV	1401.15	0.0405
15	O IV	790.112	O IV	624.618	0.0417	16	O III	833.715	O III	1666.14	0.0493
17	Si XI	582.886	Si XI	303.582	0.0540	18	Si X	621.079	Si X	287.092	0.0480
19	Si IX	692.731	Si IX	344.951	0.0263	20	Fe XV	419.552	Fe XV	396.893	0.0430
21	Fe XV	419.552	Fe XV	281.342	0.0757	22	Fe XIV	447.329	Fe XIV	334.172	0.0217
23	C IV	1548.18	C III	977.020	0.0775	24	C III	977.020	C II	1335.66	0.3115
25	Mg X	609.793	Mg IX	368.070	0.1991	26	Si III	1206.49	Si IV	1393.75	0.1563
27	N V	1238.82	O V	629.732	0.1808	28	O VI	1031.91	O V	629.732	0.1236
29	Fe XV	171.839	Fe XV	419.552	0.0708	30	O V	629.732	O IV	1401.15	0.1061

a global way. In other words, we exhibit the results by plotting the vector $\mathbf{v}$ $(= (\lambda, D^2, \chi^2))$ discussed above to help identify the optimal value of λ (λ_{opt}) which, will in turn be used for comparison of the recovered solution with that obtained using a standard Tichonov inversion. Figures 4.3 through 4.6 show these global results for the two models with first and second order smoothing for standard (top of plot) and perturbed (bottom of plot) inversions respectively.

Given the structure of the global results (shown in figures 4.3 through 4.6) the action of the inversion's optimisation process is clear; increased smoothing does create a smooth function, but one that does not necessarily enhance the recovery of the ratio pairs. Some details of individual solutions are shown in figures 4.7 through 4.10 where we can clearly observe the important role that λ plays in the optimisation. This series of figures demonstrates, for a range of smoothing parameters, the relationship between the recovered solution (solid line) of the respective model (dashed line) and the values of R_{calc} at the end of the RIT run. The right hand side of figures 4.7 through 4.10 show the behaviour of the ratio $\frac{R_{obs}}{R_{calc}}$ for all the line ratio pairs, with the 'Correlated' ratios (**b-b** errors only; #: $1 \rightarrow 22$) and the 'Uncorrelated' ratios (includes **b-f** errors also; #: $23 \rightarrow 30$) indicated by $*$ and $\bullet$ respectively. Clearly, looking at these figures in general we notice that when λ is small relative to X and D ($\ll 1$) the solution is under-constrained and is highly oscillatory and recovers the values of R_{obs} to within a few tenths of a percent. However, if λ is large relative to X and D ($\gg 1$) we see that the solution is over-constrained (and over-smoothed) and is detrimental to the recovery of the observed line ratios since the recovered $\xi(T_e)$ function no longer adequately 'fits' the data through the 'folding' of the emissivities. Similarly we observe that in the majority cases it is the uncorrelated ratios that are least well reproduced. This is evidence that equation (4.32) is a true χ^2 estimate since we would expect that the quantities with the highest uncertainties $\sigma_{l_{th}}$ will be given least 'weight' in the calculation and hence, be most poorly recovered. This is all highly analogous to the process of choosing a 'cut-off' point, in a inversion using singular value decomposition, that suppresses the eigenfunctions corresponding to small eigenvalues that was discussed in Section 2.1.3.1.

Using the values of the optimal smoothing parameter (λ_{opt}) identified in figure 4.3 to figure 4.10 we proceed by comparing the RIT with a standard Tichonov inversion for line intensities[5]. These values are collated in Table 4.2 for each test model where we have made use

[5]Only the 43 line intensities used to form the 30 line ratio values are used in this calculation. These lines can clearly be identified from Table 4.1.

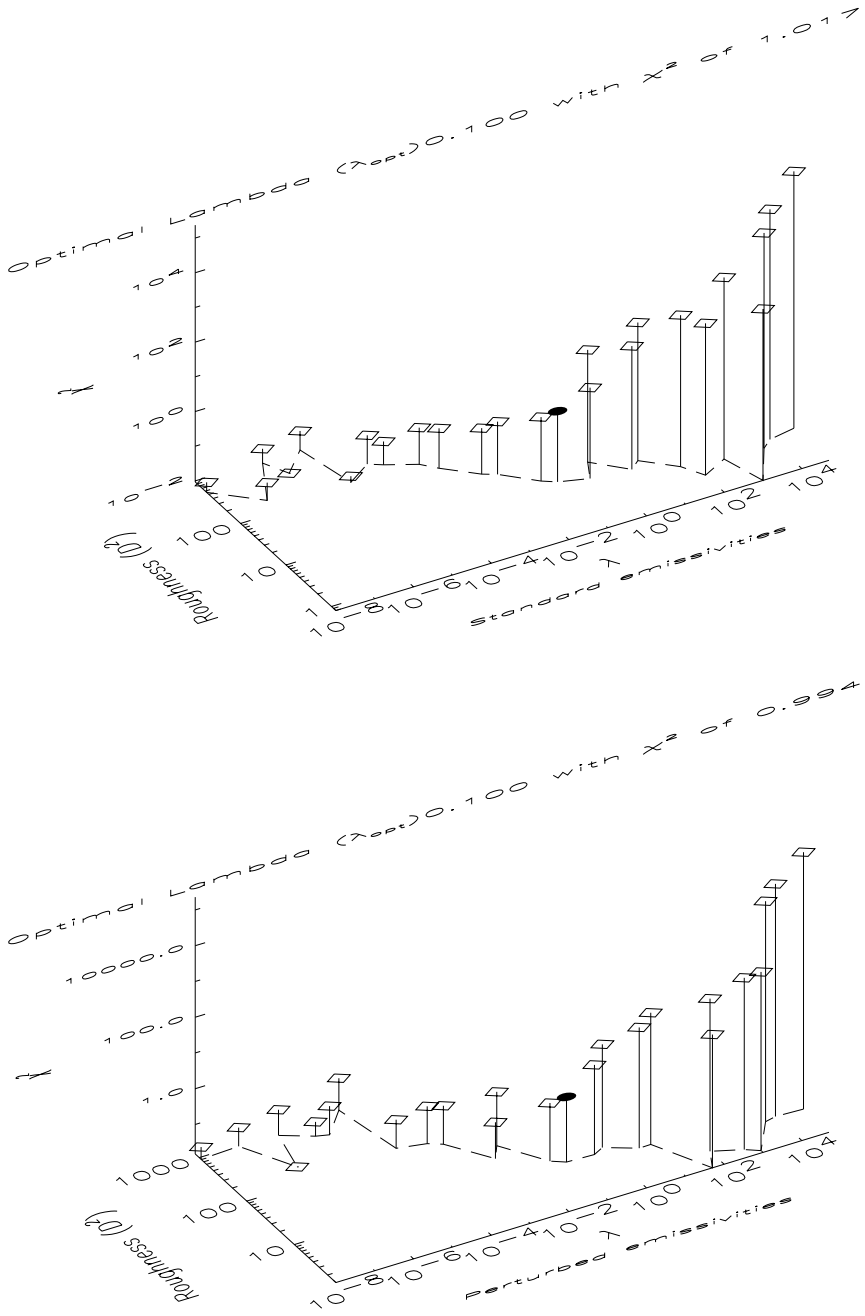

Figure 4.3: The global results of the RIT test on model 1 using a **first order** ($n = 1$) smoothing functional are best presented in this way. On plotting smoothing parameter λ versus the 'roughness' of the solution, given above as D^2 see equation (4.33) versus χ^2 calculated through equation (4.32) we are able to identify the value of λ that optimises the recovery of R_{obs} with a reasonably smooth function. This value (λ_{opt}) is determined by finding the point $\mathbf{v} = (\lambda, D^2, \chi^2)$ indicated by $\bullet$ on the upper curve closest to the origin $\mathbf{0}$, see equation (4.41). In this case $\lambda_{opt} = 0.1$ when using standard and perturbed emissivities yielding values for χ^2 of 1.017 and 0.994 respectively.

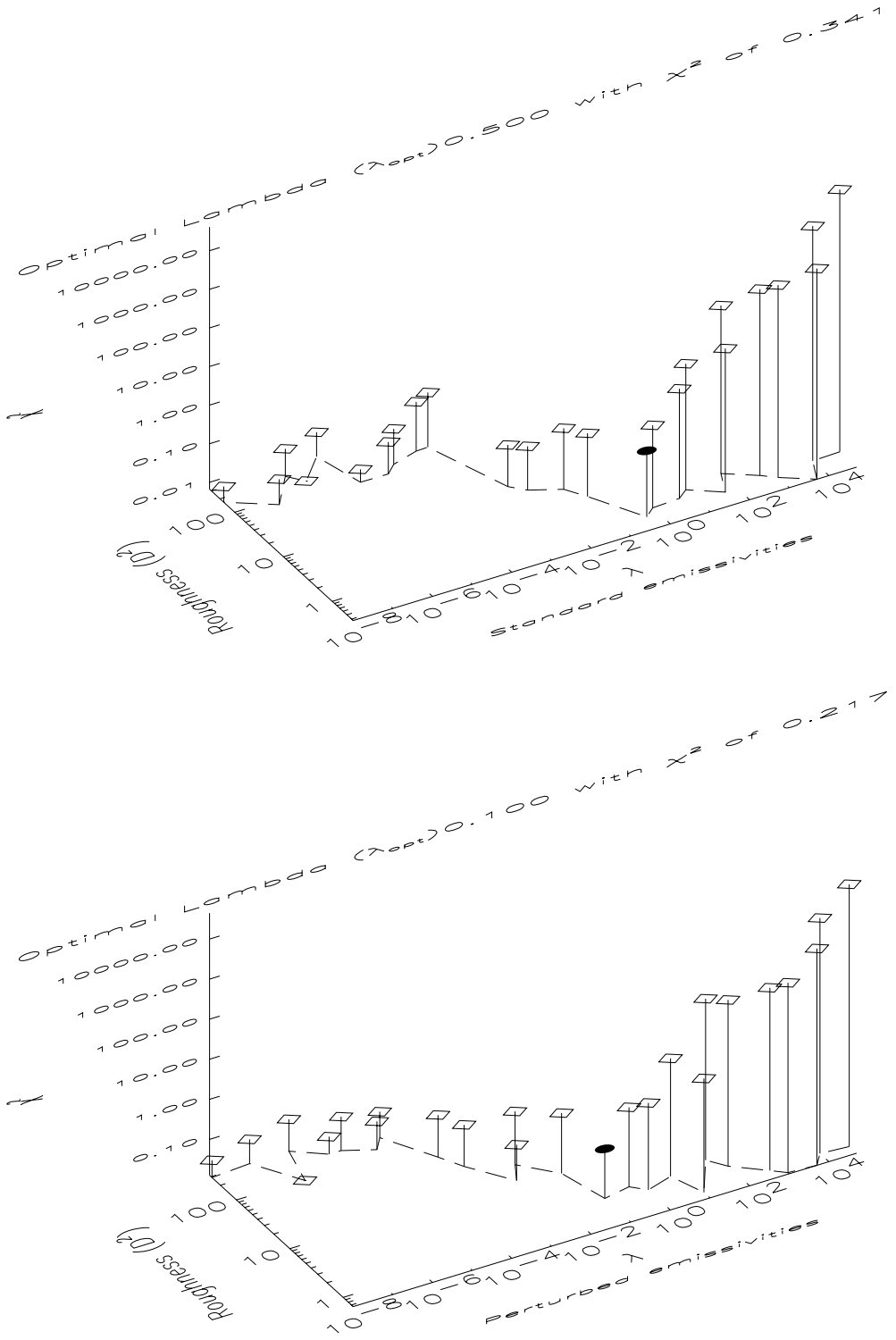

Figure 4.4: The global results of the RIT test on model 1 using a **second order** ($n = 2$) smoothing functional are presented here with quantities as described in figure 4.3. In the upper plot, for standard emissivities, λ_{opt} has a value of 0.5 which has an associated χ^2 of 0.341. Likewise, the lower plot, for perturbed emissivities, λ_{opt} has a value of 0.1 which has an associated χ^2 of 0.217. Again, λ_{opt} is indicated by $\bullet$ on the upper curve.

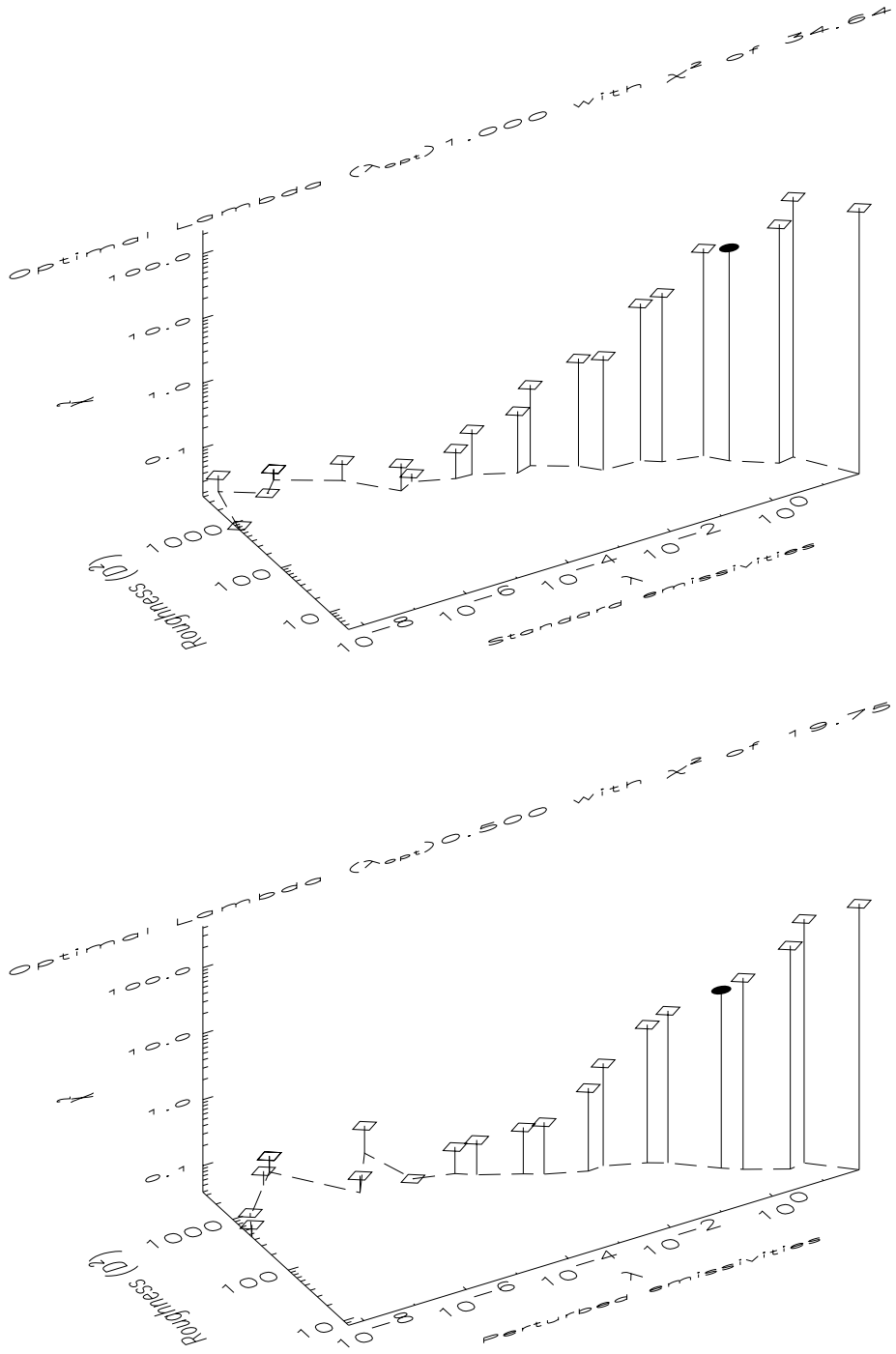

Figure 4.5: The global results of the RIT test on model 2 using a **first order** ($n = 1$) smoothing functional are presented here with quantities as described in figure 4.3. In the upper plot, for standard emissivities, λ_{opt} has a value of 1.0 which has an associated χ^2 of 34.64. Likewise, the lower plot, for perturbed emissivities, λ_{opt} has a value of 0.5 which has an associated χ^2 of 19.75. It is clear that the χ^2 calculation is dominated by the discontinuities in model 2 through the large values of D^2 associated with first order smoothing. Again, λ_{opt} is indicated by $\bullet$ on the upper curve.

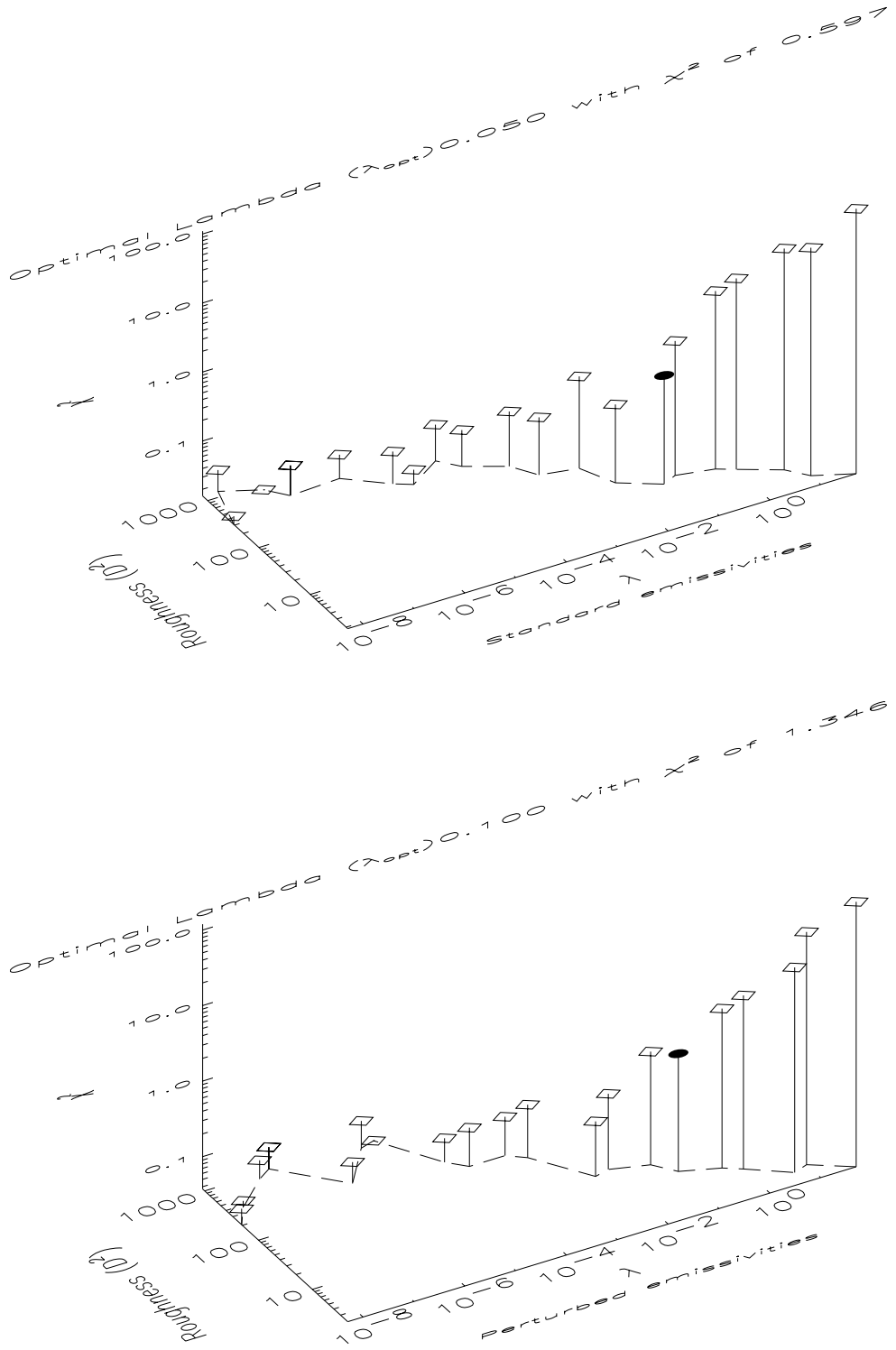

Figure 4.6: The global results of the RIT test on model 2 using a **second order** ($n = 2$) smoothing functional are presented here with quantities as described in figure 4.3. In the upper plot, for standard emissivities, λ_{opt} has a value of 0.05 which has an associated χ^2 of 0.597. Likewise, the lower plot, for perturbed emissivities, λ_{opt} has a value of 0.1 which has an associated χ^2 of 1.346. Again, λ_{opt} is indicated by • on the upper curve.

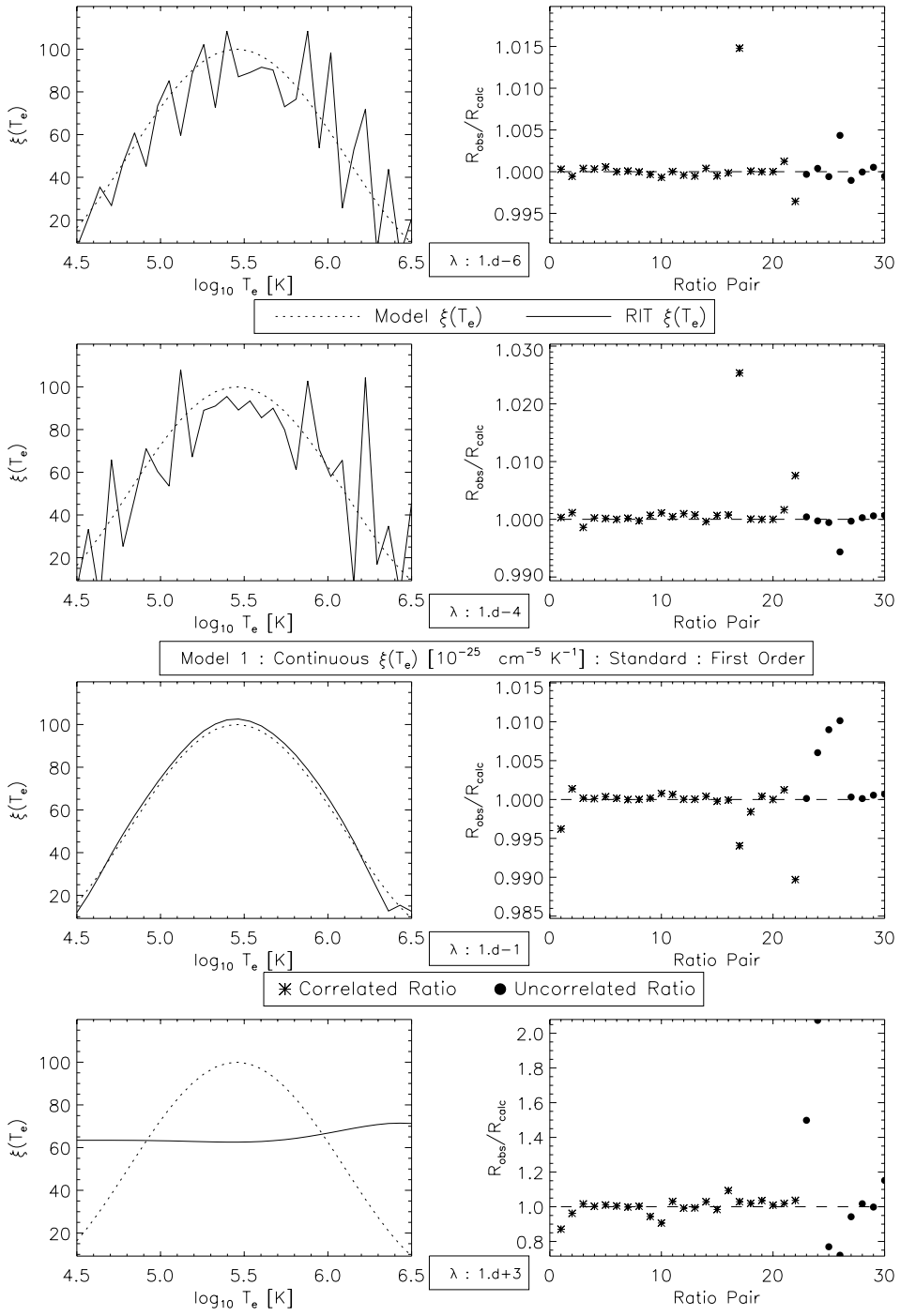

Figure 4.7: Plots showing details of single RIT runs (used to create the upper portion of figure 4.3) for test model 1 and a range of different smoothing parameters λ and a **first** order smoothing functional. The left hand plots show the solution returned by the RIT at the end of its 10,000 generation run (solid line) and the model (dashed line) for **standard** emissivities. The right hand plots demonstrate how well the actual line ratios R_{calc} for each pair are recovered. The 'Correlated' ratios (errors in **b-b** rates only) are indicated by $*$ and the 'Uncorrelated' ratios (errors in **b-f** rates also) are indicated by $\bullet$.

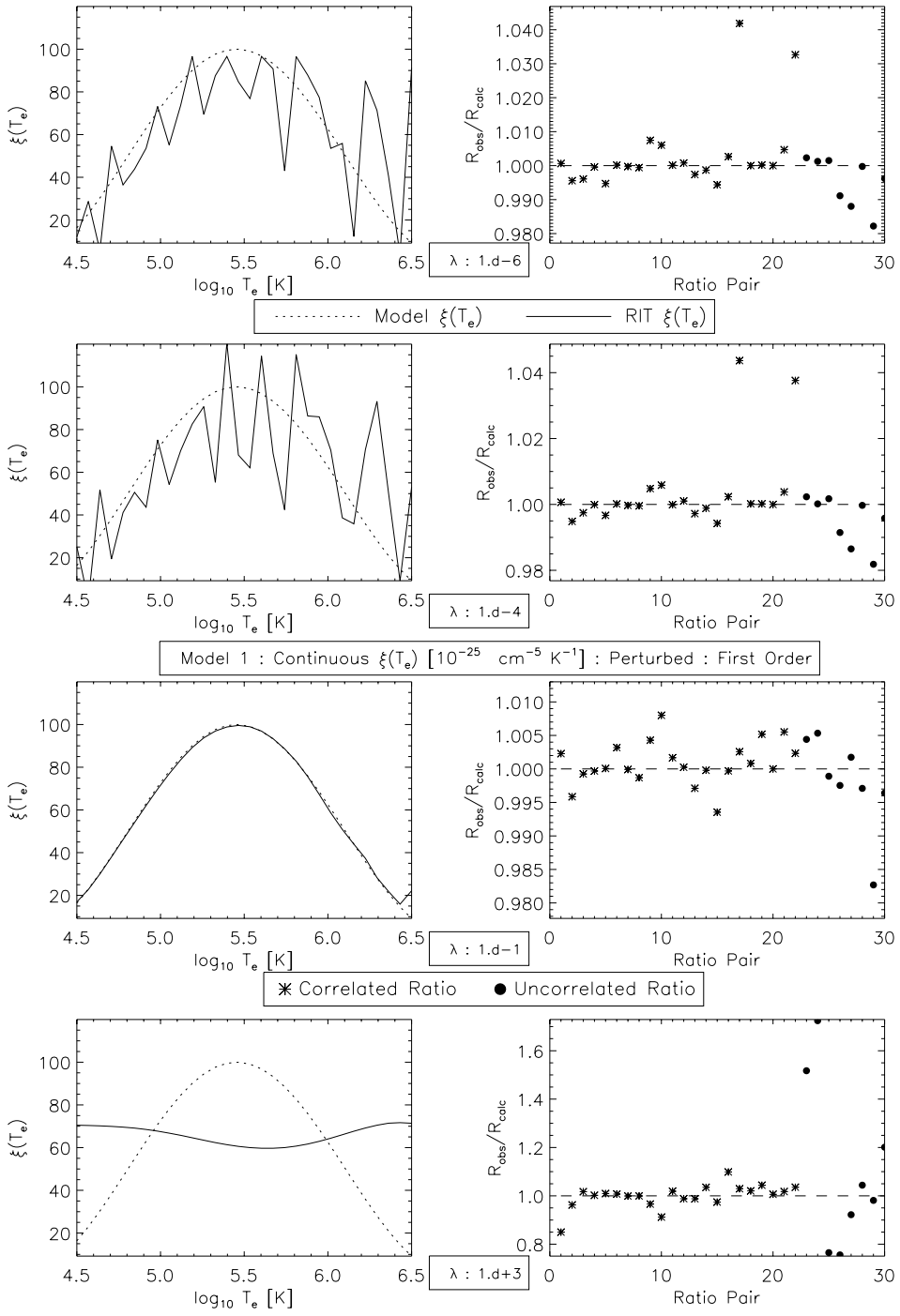

Figure 4.8: Plots showing details of single RIT runs (used to create the upper portion of figure 4.3) for test model 1 and a range of different smoothing parameters λ and a **first** order smoothing functional. The left hand plots show the solution returned by the RIT at the end of its 10,000 generation run (solid line) and the model (dashed line) for **perturbed** emissivities. The right hand plots demonstrate how well the actual line ratios R_{calc} for each pair are recovered. The 'Correlated' ratios (errors in **b-b** rates only) are indicated by $*$ and the 'Uncorrelated' ratios (errors in **b-f** rates also) are indicated by $\bullet$.

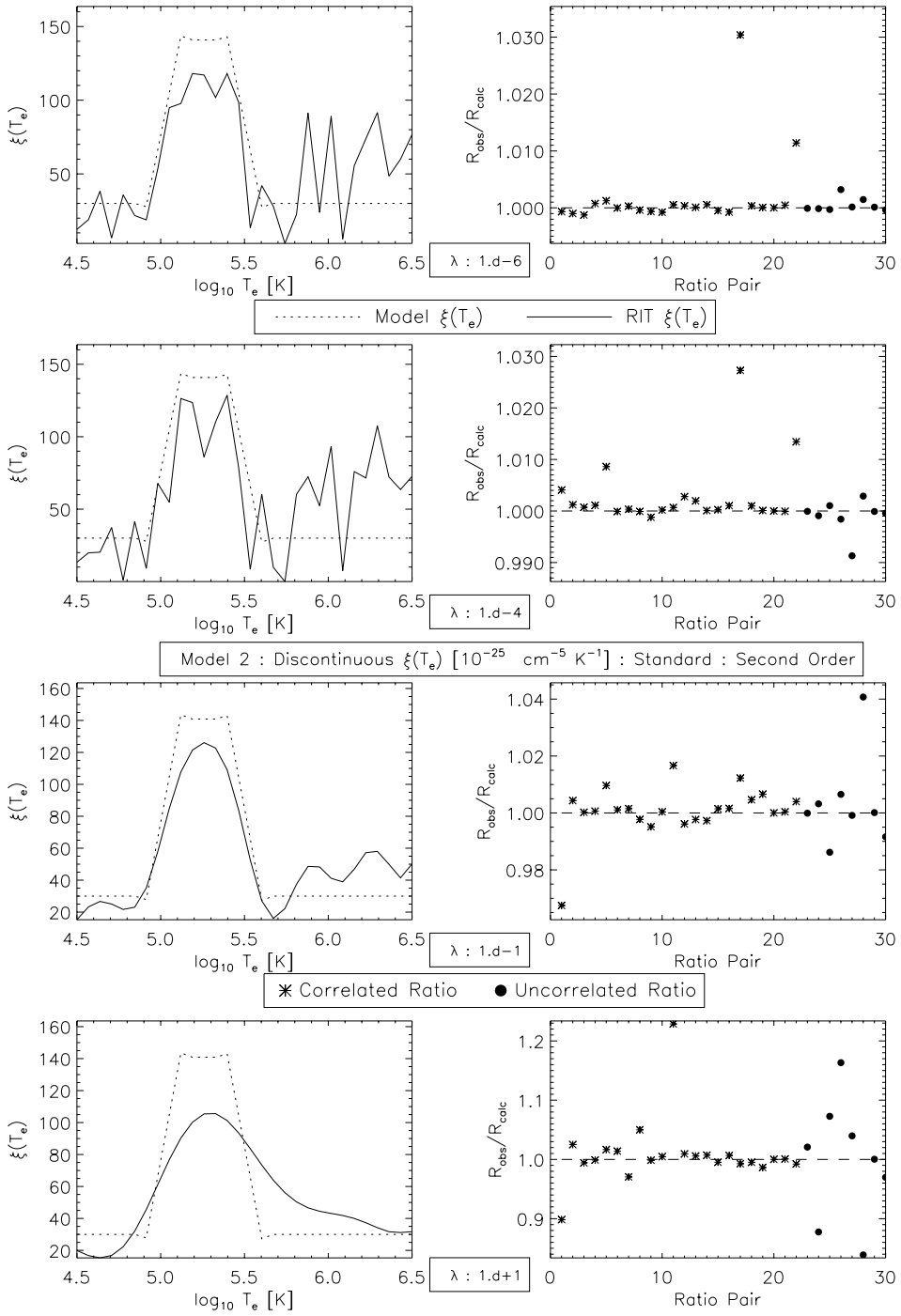

Figure 4.9: Plots showing details of single RIT runs (used to create the upper portion of figure 4.6) for test model 2 and a range of different smoothing parameters λ and a **second** order smoothing functional. The left hand plots show the solution returned by the RIT at the end of its 10,000 generation run (solid line) and the model (dashed line) for **standard** emissivities. The right hand plots demonstrate how well the actual line ratios R_{calc} for each pair are recovered. The 'Correlated' ratios (errors in **b-b** rates only) are indicated by $*$ and the 'Uncorrelated' ratios (errors in **b-f** rates also) are indicated by $\bullet$.

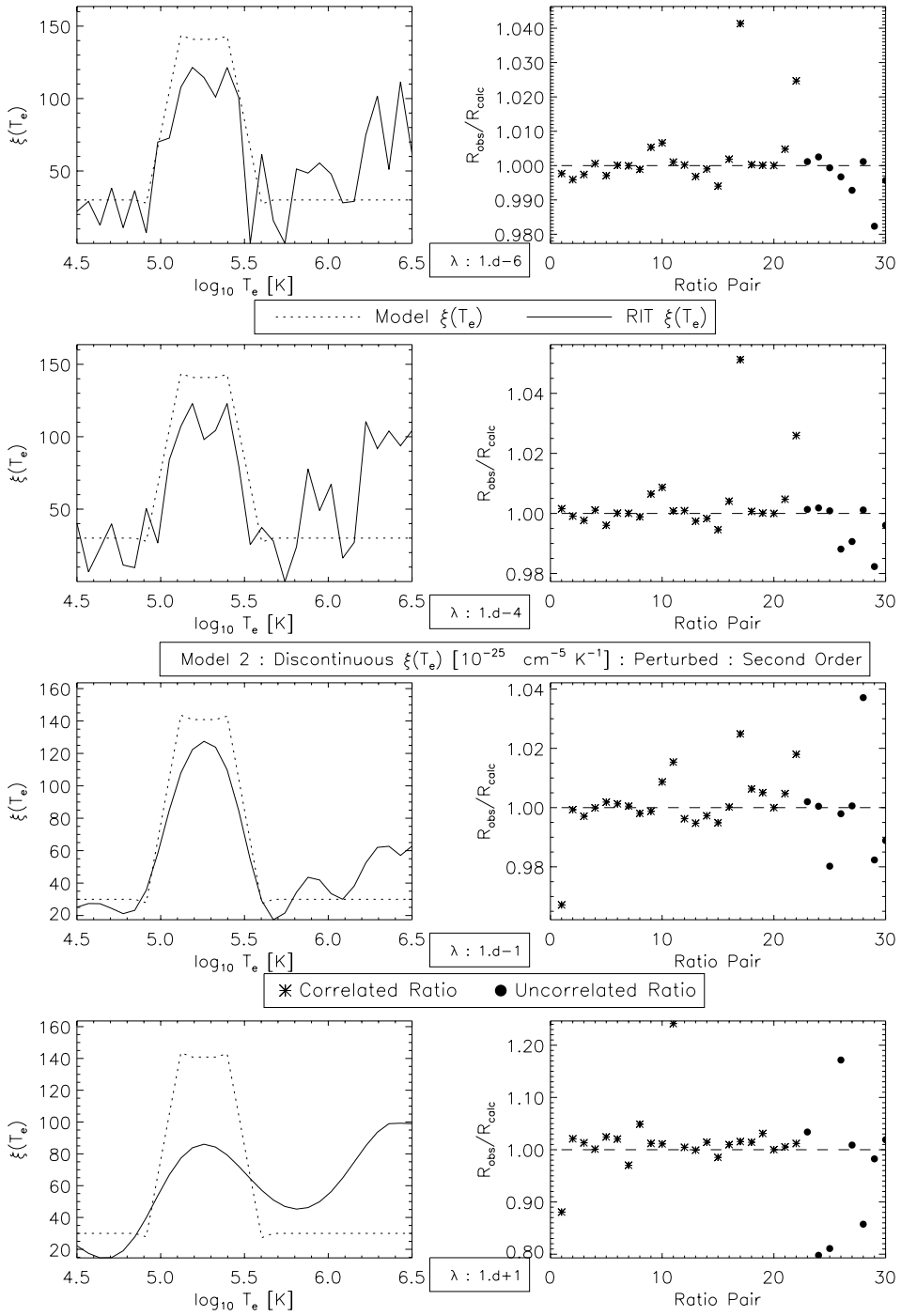

Figure 4.10: Plots showing details of single RIT runs (used to create the upper portion of figure 4.6) for test model 2 and a range of different smoothing parameters λ and a **second** order smoothing functional. The left hand plots show the solution returned by the RIT at the end of its 10,000 generation run (solid line) and the model (dashed line) for **perturbed** emissivities. The right hand plots demonstrate how well the actual line ratios R_{calc} for each pair are recovered. The 'Correlated' (errors in **b-b** rates only) ratios are indicated by $*$ and the 'Uncorrelated' ratios (errors in **b-f** rates also) are indicated by $\bullet$.

of equation (4.39) to estimate such an optimal value for the Tichonov inversion. This inversion comparative is anticipated to show that the perturbations applied to the line emissivities can be catered for in the RIT but *not* in a standard routine by design. Indeed, we present the set of comparative test results for the RIT in figures 4.11 and 4.12. These figures, as stated above, demonstrate the effectiveness of the RIT in combatting the effects imposed on the inversion of equation (4.28) by perturbations in the line emissivities compared to that of a the standard line intensity inversions. From these figures it is clear that although the RIT provides a more than adequate inversion for both of the test models when using the standard emissivities it really does come into its own when supplied with the randomly selected set of perturbed emissivities. Indeed, the latter is a very appropriate test since, in many situations when analysing remotely sensed UV spectra, we cannot be sure about the nature of the emitting plasma to appropriately define the peculiarities of the line emissivities required to perform the inversion.

Table 4.2: Details of optimal values of smoothing parameter λ extracted from the various the RIT runs on the two test models for the different smoothing functionals. These values are principally taken from figure 4.3 to figure 4.7.

Model Number	Smoothing Order	Perturbed / Standard	λ_{opt}^{RIT}	$\log_{10}\ \lambda_{opt}^{TICH}$
1	1^{st}	Standard	0.10	5.30
1	1^{st}	Perturbed	0.10	6.30
1	2^{nd}	Standard	0.50	5.25
1	2^{nd}	Perturbed	0.10	5.80
2	1^{st}	Standard	1.00	6.00
2	1^{st}	Perturbed	0.50	6.05
2	2^{nd}	Standard	0.05	5.45
2	2^{nd}	Perturbed	0.10	5.75

So, in terms of obtaining unique 'reliable' inversions of solar UV spectroscopic data to obtain useful empirical and physical models of the emitting structure, the RIT shows that is a more than capable alternative to standard regularised inversions and the figures included here essentially speak for themselves especially when we note that the RIT explicitly *forces* its solutions to be strictly positive which is not always the case for standard inversions.

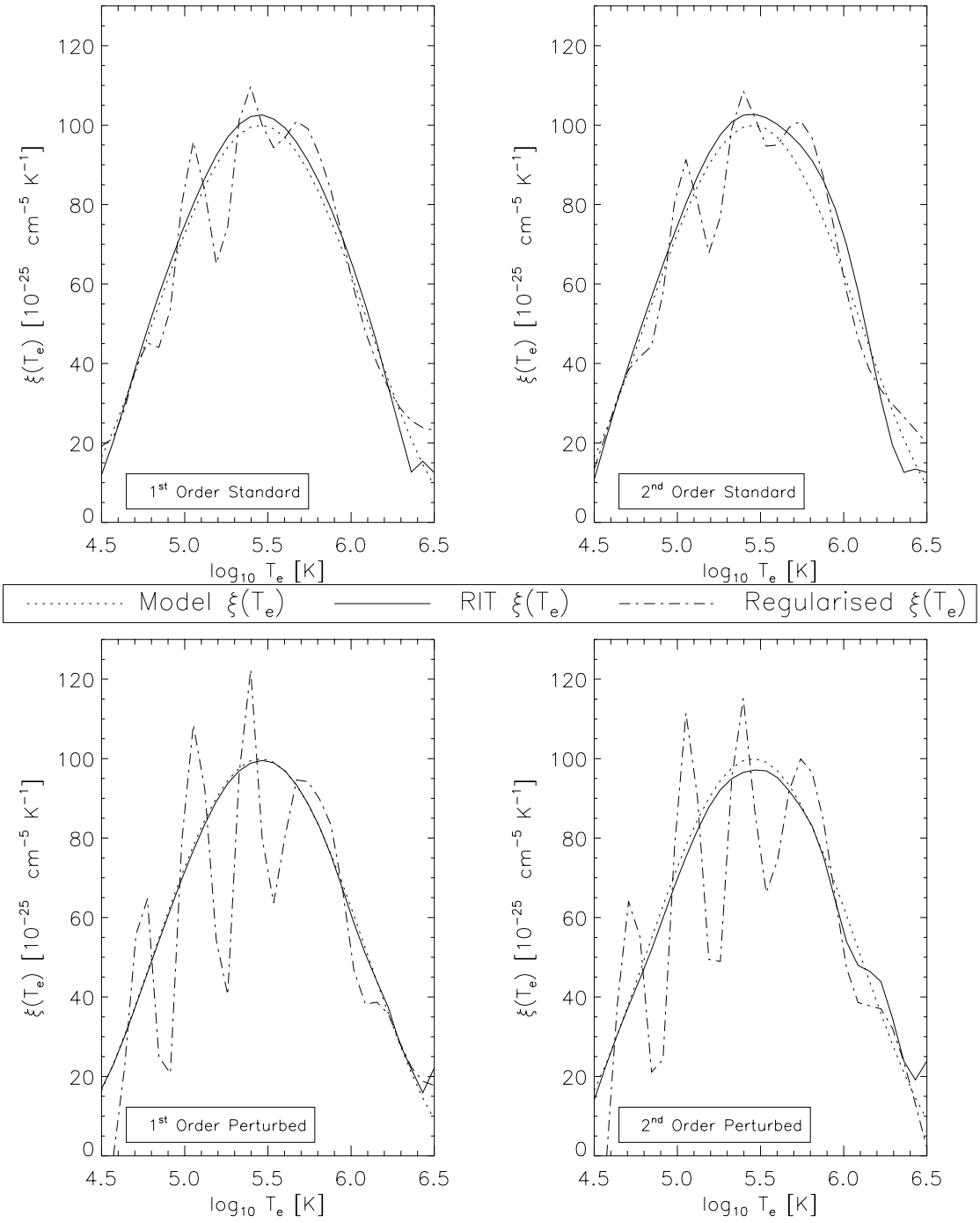

Figure 4.11: Comparative results for the RIT and a standard inversion when using both standard (upper plots) and perturbed (lower plots) emissivities. Recovered functions from the RIT (solid lines) and a standard Tichonov regularisation (dot-dash lines) are plotted against the test model; in this case model 1, the continuous test model. The values of λ_{opt} can be obtained from Table 4.2.

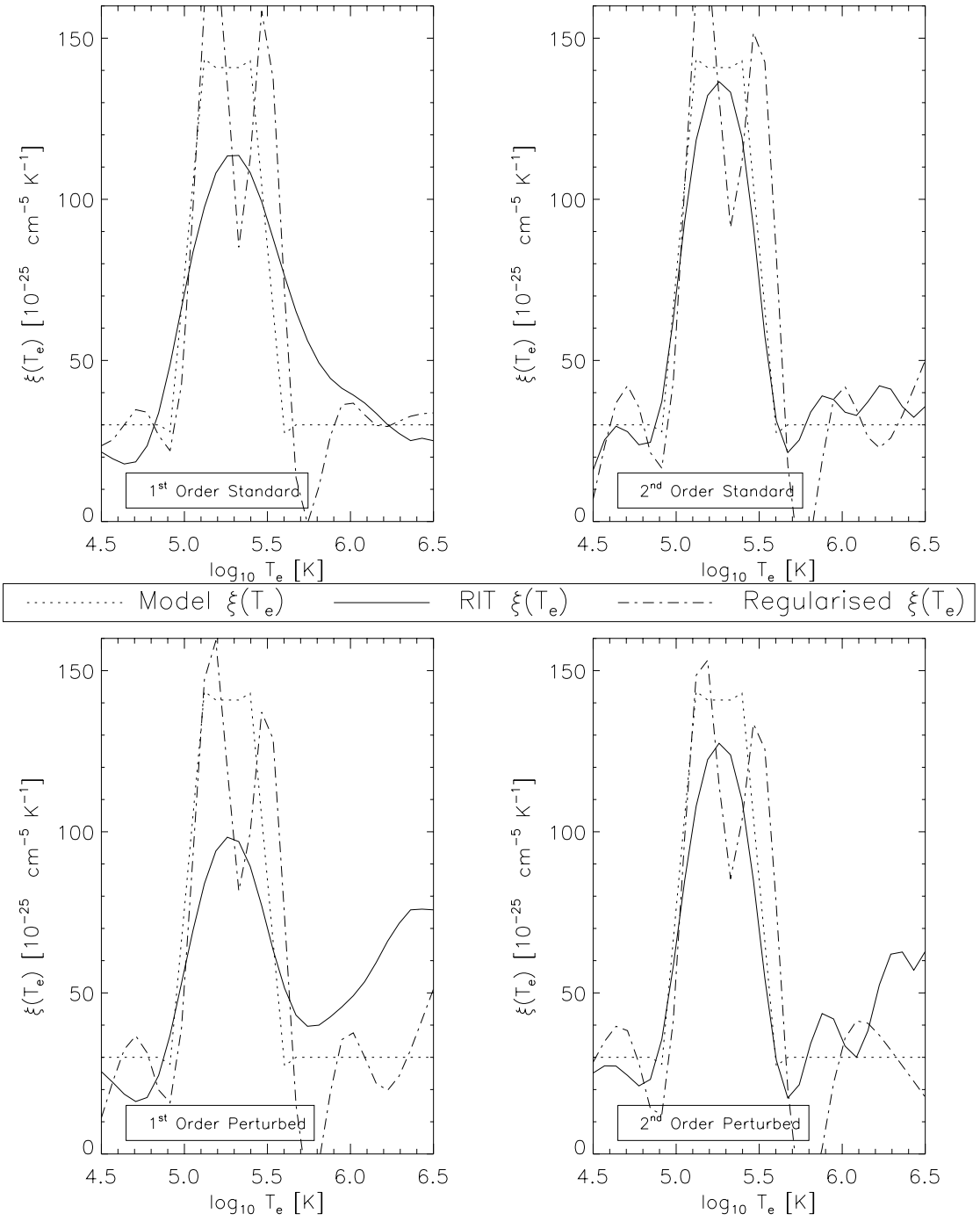

Figure 4.12: Comparative results for the RIT and a standard inversion when using both standard (upper plots) and perturbed (lower plots) emissivities. Recovered functions from the RIT (solid lines) and a standard Tichonov regularisation (dot-dash lines) are plotted against the test model; in this case model 1, the continuous test model. The values of λ_{opt} can be obtained from Table 4.2.

4.3.2 RIT test results for $\zeta(n_e)$

In a completely analogous approach to that immediately above we test the RIT in a bid to recover the differential emission measure in electron density $\zeta(n_e)$ from a series of synthetically generated line ratios. Here we only present the results of the RIT for one test model form of $\zeta(n_e)$ because it is a diagnostic of the emitting plasma not commonly discussed in the literature. It is anticipated that any form of inversion for $\zeta(n_e)$, whether it be using the RIT or a standard regularisation routine, will suffer from serious problems associated with the poor conditioning (see, e.g., Section 2.1.2) of the set of line emissivities used. Again we leave discussion of this effect to Chapter 5. A gross simplification is that, where the line emissivities of the optically thin emission lines are relatively peaked functions in T_e and are *not* so when considered as functions of n_e. One look at figures 5.1 and 5.9 will convince the reader of that. As a consequence of the functional nature of the line emissivities the condition number C_K of equation (4.29) is much larger than that of the $\xi(T_e)$ inversion case above and the degree of numerical stability in the inversion is dramatically reduced. Hence, the solutions are *more* sensitive to data noise and are likely to be highly oscillatory in nature.

So, neglecting the issues concerning the poor conditioning we present the RIT test results for a test model (and problem) that is conceptually no different from those presented above. Here the test model is a 'Step' function over the n_e domain ($8.5 \leq \log_{10}\ n_e \leq 12.5$) and is shown in figure 4.13. Again, we have performed the 'forward-backward' analysis described in Section 4.3 with the line intensities, ratios, standard and distribution of 20 perturbed emissivities calculated using the recipe of Section 4.2.1.

Table 4.3 identifies the line ratio pairs l used in these calculations along with their wavelengths (λ Å) and their measure of the theoretical uncertainty in the line ratio ϵ_l ($\sigma_{l_{th}}$ as a fraction of $R_{l_{th}}$ for a flat model $\zeta(n_e)$ function). It is clear that, as anticipated, the line ratio pairs with each line belonging to a common ionisation stage of the atom having considerably lower values, in general, than others being typically in the range of $\epsilon_l \approx 2 - 6\%$. The 'Uncorrelated' line ratio pairs have typical values averaging around 10%.

The recovered form of $\zeta(n_e)$ from the various runs using different smoothing orders and sets of emissivities are, as above, best considered globally before extracting specific optimal solutions for comparison with the standard line intensity approach. So, we again plot the vector $\mathbf{v}$ ($= (\lambda, D^2, \chi^2)$) for each different RIT setup to identify the optimal value of λ (λ_{opt}). Figures 4.14 and 4.15 show these plots for the $\zeta(n_e)$ step function with first and second order

Table 4.3: Details of the line pairs used in the RIT runs on $\zeta(n_e)$ presented in this chapter. For each ratio pair $l = (i, j)$ of R_{ij} the numerator, i, (N) and denominator, j, (D) lines are indicated, along with the ionic stage to which they belong and their wavelength (λ Å). Also quoted is the measure of uncertainty ϵ_l (i.e. σ_{th_l} as a percentage of the theoretical line ratio R_{th_l} for a flat model DEM) from the distribution of 20 perturbed line emissivities. Ratio pairs 1 through 18 are known here as 'Correlated' ratios since they have errors in the **b-b** rates only whereas pairs 19 through 24 are 'Uncorrelated' and include errors in the **b-f** rates also. Note also that some of the ratio pairs contain *no* density information, e.g., pair #1 is the ratio of two resonance lines.

#	Ion_N	λ_N	Ion_D	λ_D	ϵ_l	#	Ion_N	λ_N	Ion_D	λ_D	ϵ_l
1	C IV	1548.18	C IV	312.420	0.0398	2	C III	977.020	C III	1175.26	0.0533
3	Mg IX	706.060	Mg IX	368.070	0.0376	4	Mg IX	706.060	Mg IX	445.980	0.0261
5	Ne VII	895.175	Ne VII	465.220	0.0375	6	Ne VII	895.175	Ne VII	562.993	0.0497
7	Ne VI	562.711	Ne VI	999.630	0.0550	8	Ne VI	562.711	Ne VI	454.072	0.0861
9	Si III	1206.49	Si III	1301.14	0.0638	10	N III	991.502	N III	772.385	0.0283
11	O VI	1031.91	O VI	150.089	0.0227	12	O V	1218.34	O V	629.732	0.0340
13	O V	1218.34	O V	761.128	0.0555	14	O IV	790.112	O IV	1401.15	0.0334
15	O IV	790.112	O IV	624.618	0.0414	16	O III	833.715	O III	1666.14	0.0386
17	Si X	621.079	Si X	287.092	0.0335	18	Si IX	692.731	Si IX	344.951	0.0429
19	C IV	1548.18	C III	977.020	0.0533	20	C III	977.020	C II	1335.66	0.0675
21	Si III	1206.49	Si IV	1393.75	0.0588	22	N V	1238.82	O V	629.732	0.2347
23	O VI	1031.91	O V	629.732	0.0359	24	O V	629.732	O IV	1401.15	0.1528

smoothing for standard (top of plot) and perturbed (bottom of plot) emissivity inversions respectively. Similarly to figures 4.7 through 4.10 we present details of some specific runs of the RIT in figures 4.16 and 4.17 which clearly demonstrate the relationship between the recovered solution (solid line) of the 'Step' model (dashed line) and the values of R_{calc} at the end of the RIT run. Again, the right hand side of the figures show the behaviour of the ratio $\frac{R_{obs}}{R_{calc}}$ for all the line ratio pairs, the 'Correlated' ratios ($\# : 1 \rightarrow 18$) and the 'Uncorrelated' ratios ($\# : 19 \rightarrow 24$) which are indicated by $*$ and $\bullet$ respectively.

Table 4.4: Details of optimal values of smoothing parameter λ extracted from the various the RIT runs on the single test model for the different smoothing functionals. These values are principally taken from figures 4.14 and 4.15.

Smoothing Order	Perturbed / Standard	λ_{opt}^{RIT}	$\log_{10}\ \lambda_{opt}^{TICH}$
1^{st}	Standard	0.010	4.20
1^{st}	Perturbed	0.001	4.40
2^{nd}	Standard	0.010	4.00
2^{nd}	Perturbed	0.001	4.10

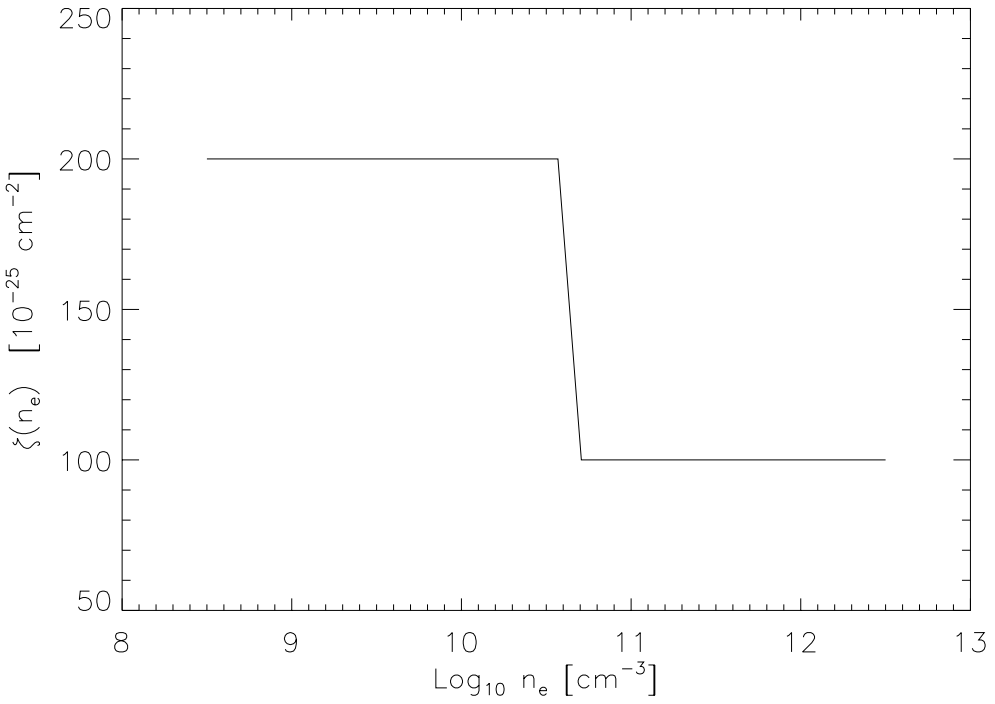

Figure 4.13: Plot of the 'Step' test model of $\zeta(n_e)$.

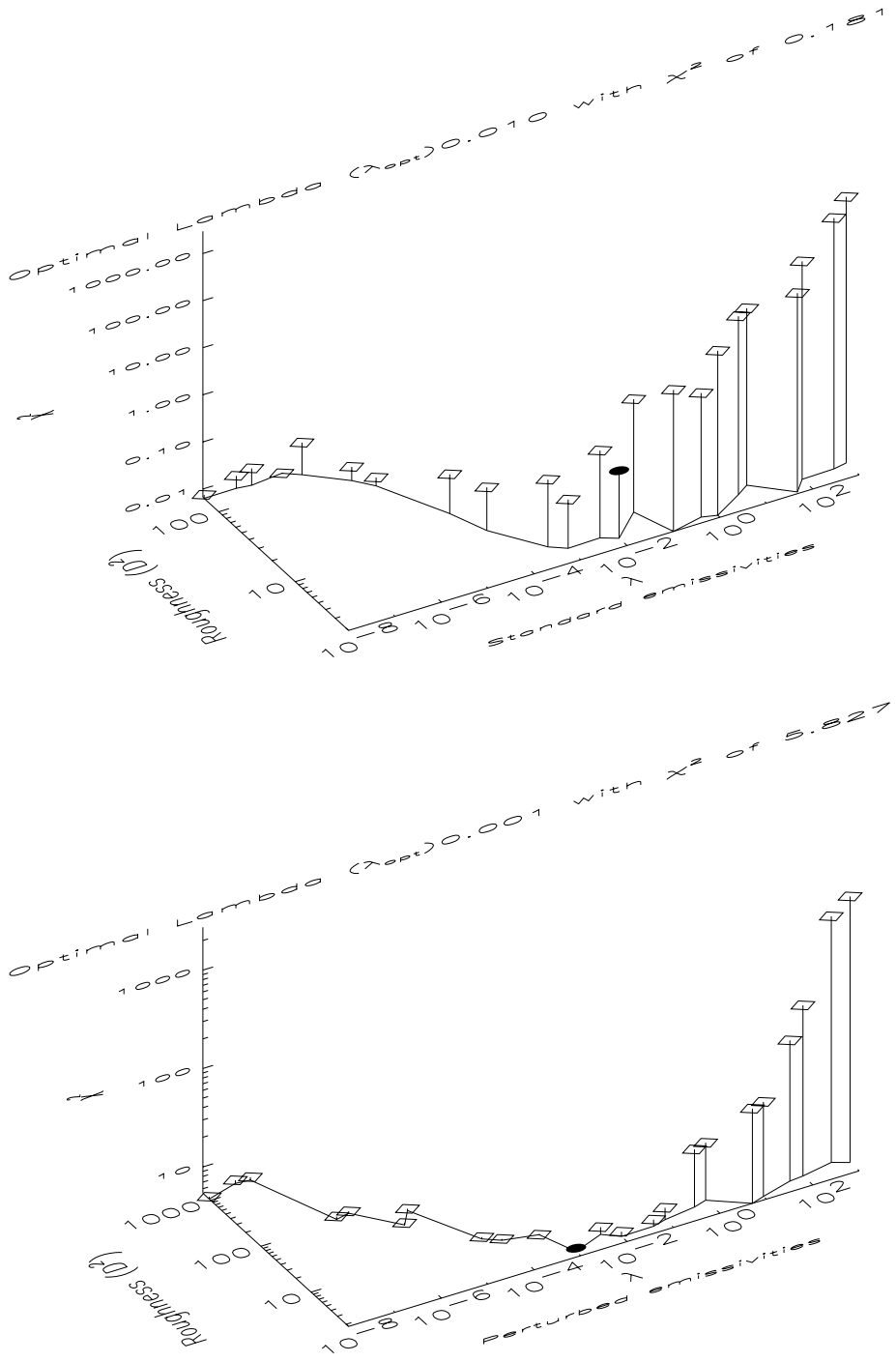

Figure 4.14: The global results of the RIT test on the 'Step' model for $\zeta(n_e)$ using a **first order** ($n = 1$) smoothing functional are presented here with quantities as described above. In the upper plot, for standard emissivities, λ_{opt} has a value of 0.01 which has an associated χ^2 of 0.181. Likewise, the lower plot, for perturbed emissivities, λ_{opt} has a value of 0.001 which has an associated χ^2 of 5.827. As in previous figures it is clear that the χ^2 calculation is dominated by the discontinuities in the model. Again, λ_{opt} is indicated by $\bullet$ on the upper curve.

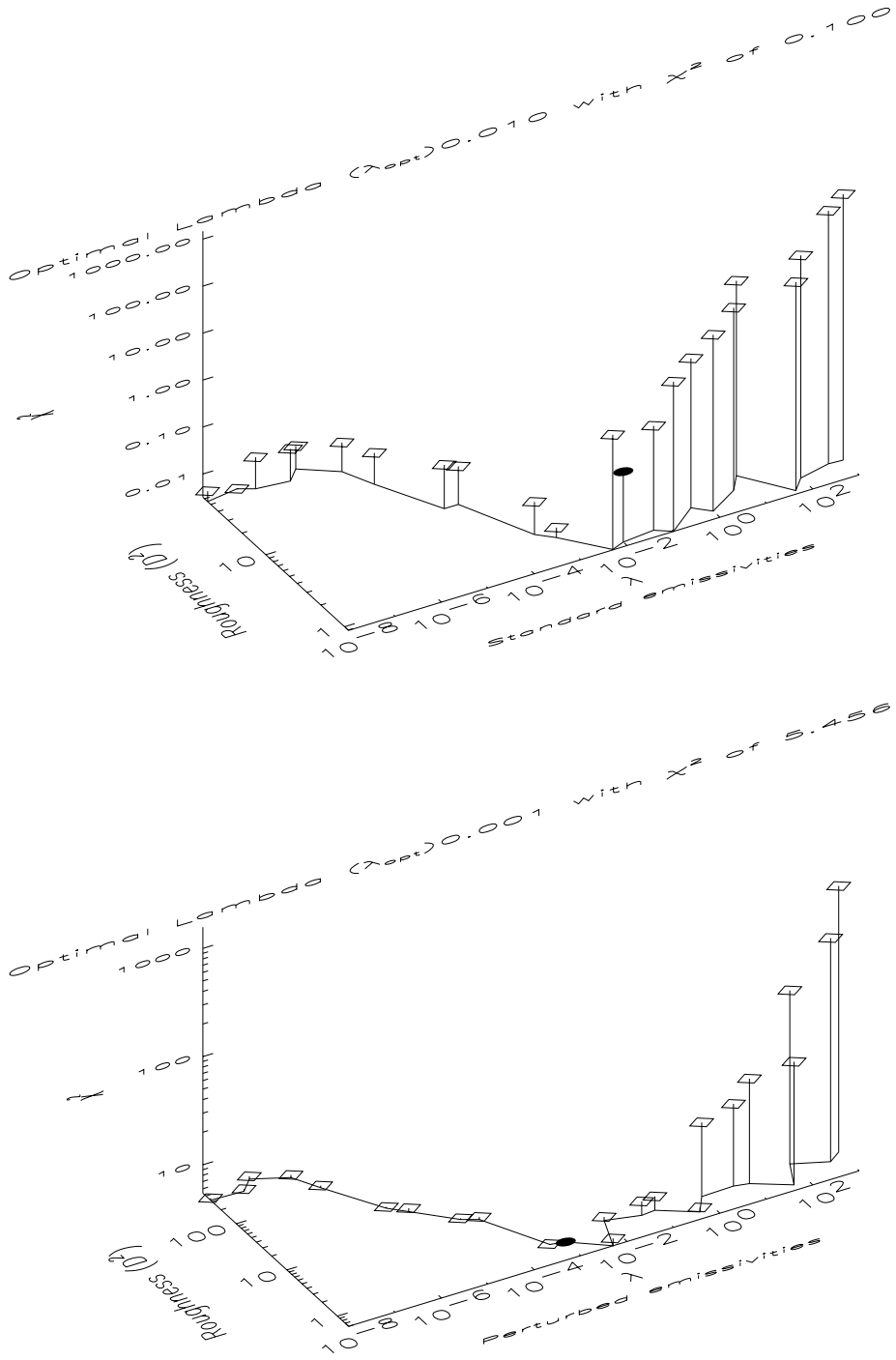

Figure 4.15: The global results of the RIT test on the 'Step' model for $\zeta(n_e)$ using a **first order** ($n = 2$) smoothing functional are presented here with quantities as described above. In the upper plot, for standard emissivities, λ_{opt} has a value of 0.01 which has an associated χ^2 of 0.100. Likewise, the lower plot, for perturbed emissivities, λ_{opt} has a value of 0.001 which has an associated χ^2 of 5.456. As in previous figures it is clear that the χ^2 calculation is dominated by the discontinuities in the model. Again, λ_{opt} is indicated by $\bullet$ on the upper curve.

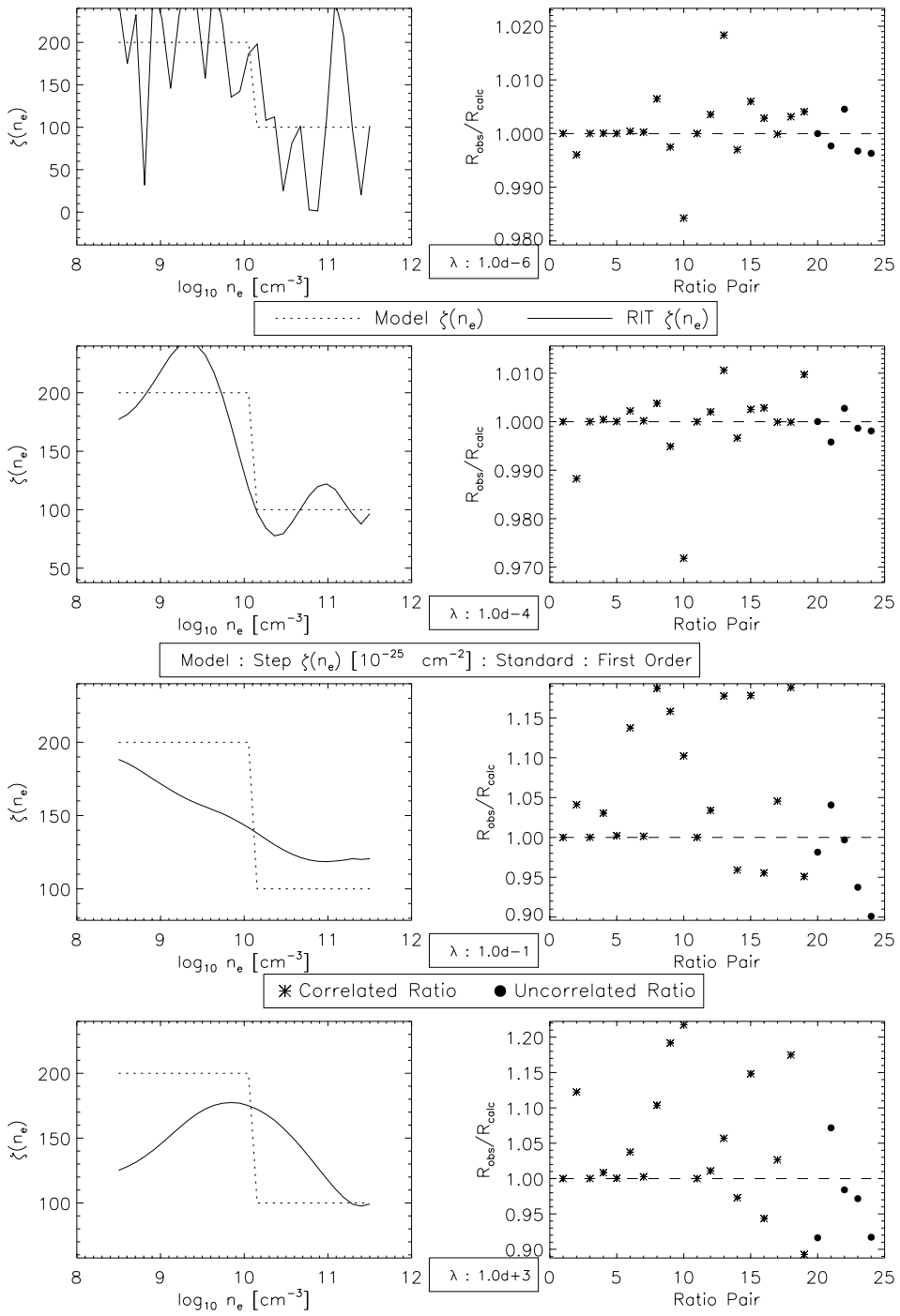

Figure 4.16: Plots showing details of single RIT runs (used to create the lower portion of figure 4.14) for the 'Step' model and a range of different smoothing parameters λ and a **first** order smoothing functional. The left hand plots show the solution returned by the RIT at the end of its 10,000 generation run (solid line) and the model (dashed line) for **standard** emissivities. The right hand plots demonstrate how well the actual line ratios R_{calc} for each pair are recovered. The 'Correlated' ratios (errors in **b-b** rates) are indicated by $*$ and the 'Uncorrelated' ratios (errors in **b-f** rates also) are indicated by $\bullet$.

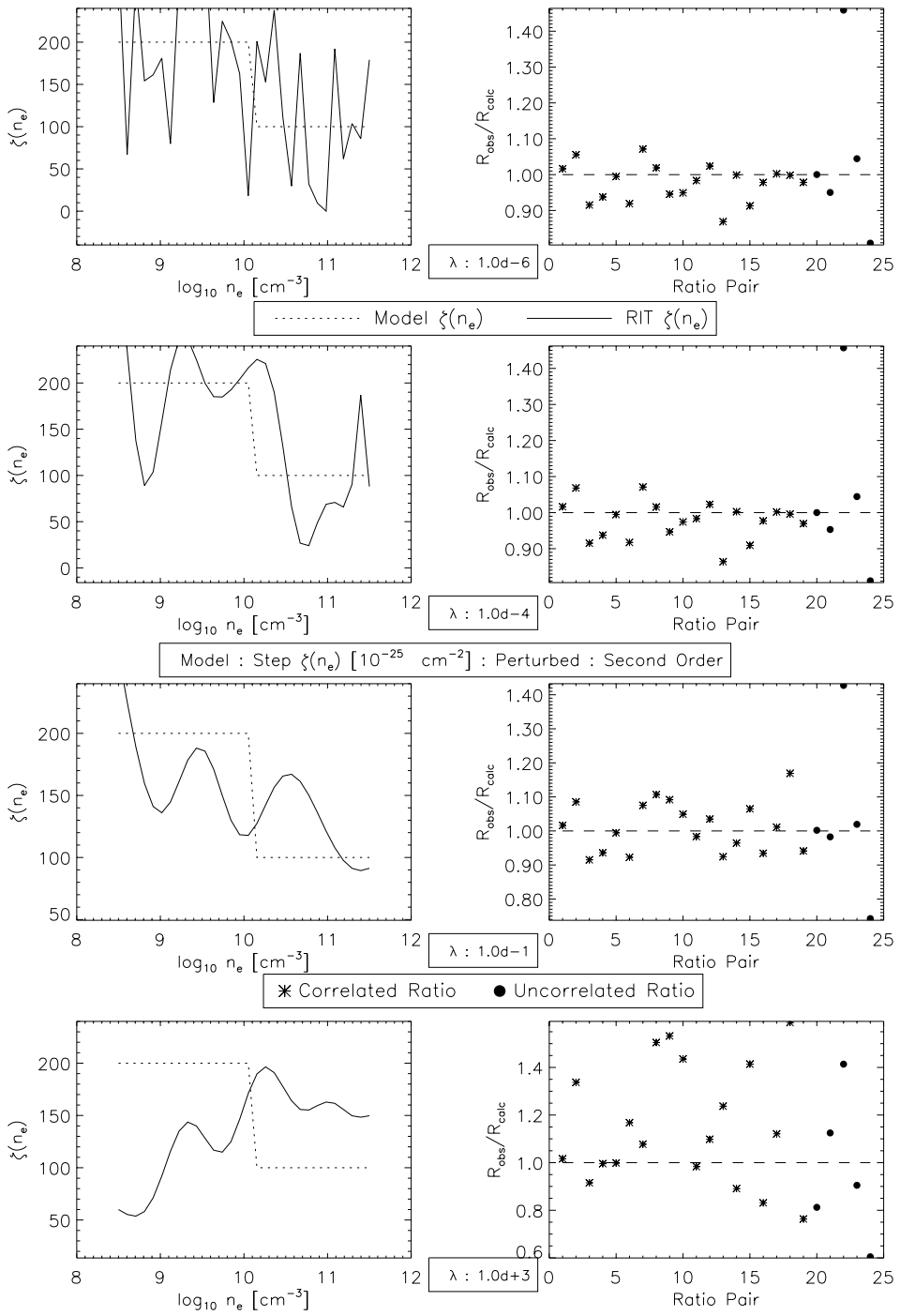

Figure 4.17: Plots showing details of single RIT runs (used to create the lower portion of figure 4.15) for the 'Step' model and a range of different smoothing parameters λ and a **second** order smoothing functional. The left hand plots show the solution returned by the RIT at the end of its 10,000 generation run (solid line) and the model (dashed line) for **perturbed** emissivities. The right hand plots demonstrate how well the actual line ratios R_{calc} for each pair are recovered. The 'Correlated' ratios (errors in **b-b** rates) are indicated by $*$ and the 'Uncorrelated' ratios (errors in **b-f** rates also) are indicated by $\bullet$.

From figure 4.18, the results of the RIT runs for this particular test, it is clear that although the RIT produces an slightly over-smoothed solution it yields a recovered structure that is significantly better than the highly oscillatory (*not strictly positive*) solution of the standard intensity inversion in the presence of 15% data noise[6]. Recalling that we prescribe only that the solution satisfy the two criteria (recovery of R_{obs} with a certain degree of prescribed smoothness, however defined) we observe little of the oscillation in the RIT test results even though the value of C_K is significantly higher than that of the previous $\xi(T_e)$ inversions (10^{17} compared to 10^{11}). It is clear again that the RIT also preserves the positivity of the recovered solutions even though the solutions appear over-smoothed this *is* an artifact of the 'flatness' of the line emissivities themselves. This effect is explained in the next chapter.

4.3.3 RIT inversion results using a generalised smoothing functional

From some of the results presented above, specifically model 2 of Section 4.3.1, are reasonable solutions given the limited nature of the smoothing functionals used. This is especially true when discontinuities in the DEM functions are likely to be present. The following discussion shows that it is *very* simple to implemented a generalised smoothing functional to resolve sharp features in DEM functions $f(s_e)$ with the RIT.

In this case we will implement a form of Maximum Entropy (ME) smoothing discussed in Section 2.1.3.3. The form of $\Phi(f(s_e))$ (cf. equation (4.33)) now being, for $f(s_e)$ discretised over N points in the s_e domain

$$\Phi(f(s_e)) = \sum_{i=1}^{N} \left(\frac{f_i}{y}\right) \ln\left(\frac{f_i}{y}\right) \tag{4.42}$$

where f_i is the evaluation of $f(s_e)$ at index point i and y is the *prior* of the solution taken to be the average summed over the entire domain ($y = \langle f(s_e) \rangle$).

The global results of the RIT runs using this ME form for the smoothing functional on Model 2 for standard and perturbed emissivities over a range of smoothing parameters are presented in figure 4.19. Again we can identify, for the perturbed emissivity inversion alone, the optimal value λ (0.580) and we use the corresponding solution to compare with the RIT recovered results for regularising functionals $n = 1, 2$. Figure 4.20 shows that using an ME approach allows a less subjective interpretation of the data; simply because no strict

[6] These oscillations in the solution are visibly less when the data noise is not so high. This can be observed for 5% intensity noise in the next chapter.

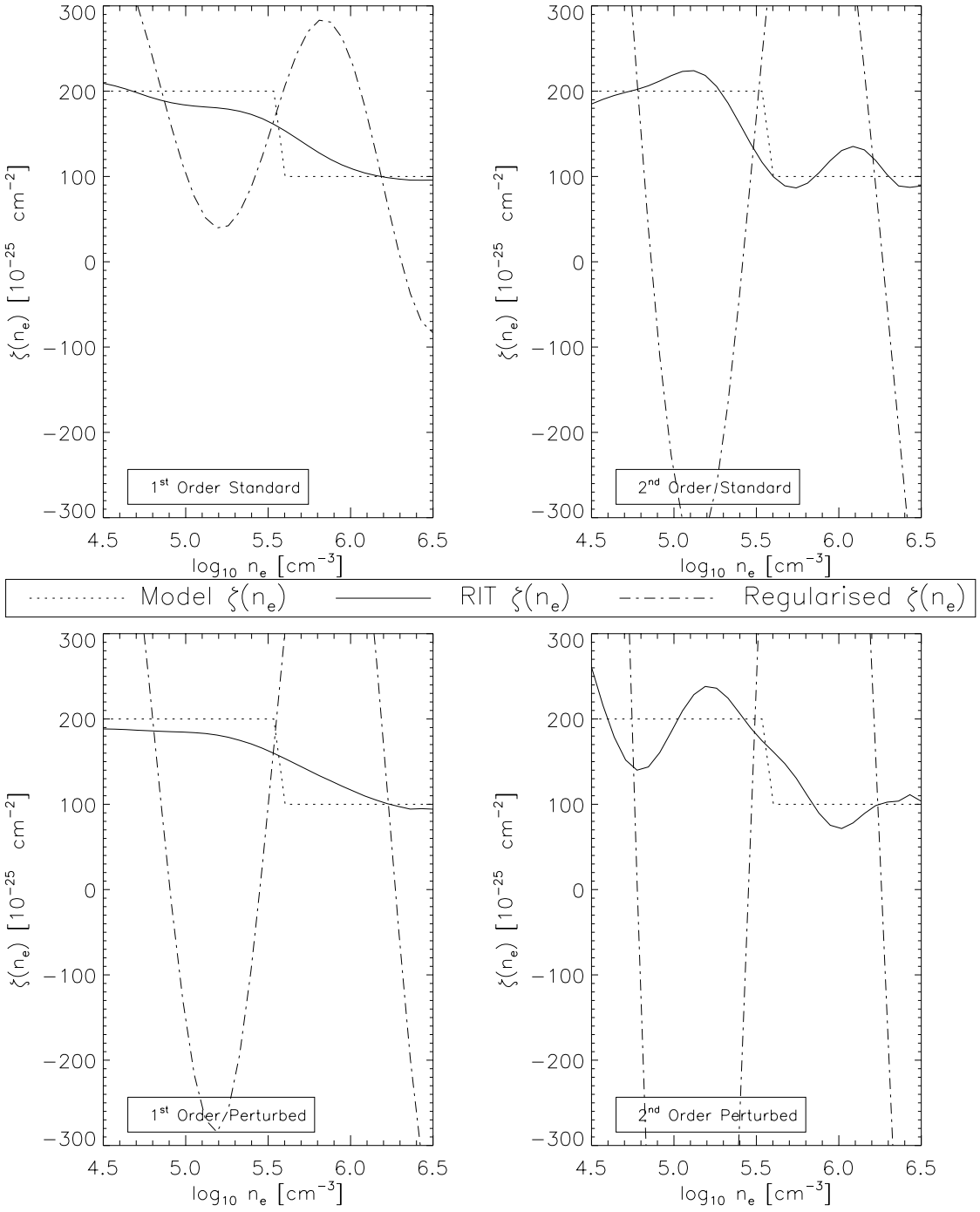

Figure 4.18: Comparative results for the RIT and a standard inversion when using both standard (upper plots) and perturbed (lower plots) emissivities. Recovered functions from the RIT (solid lines) and a standard Tichonov regularisation (dot-dash lines) are plotted against the test model; in this case model 1, the continuous test model. The values of λ_{opt} can be obtained from Table 4.4.

functional form is imposed on the solution.

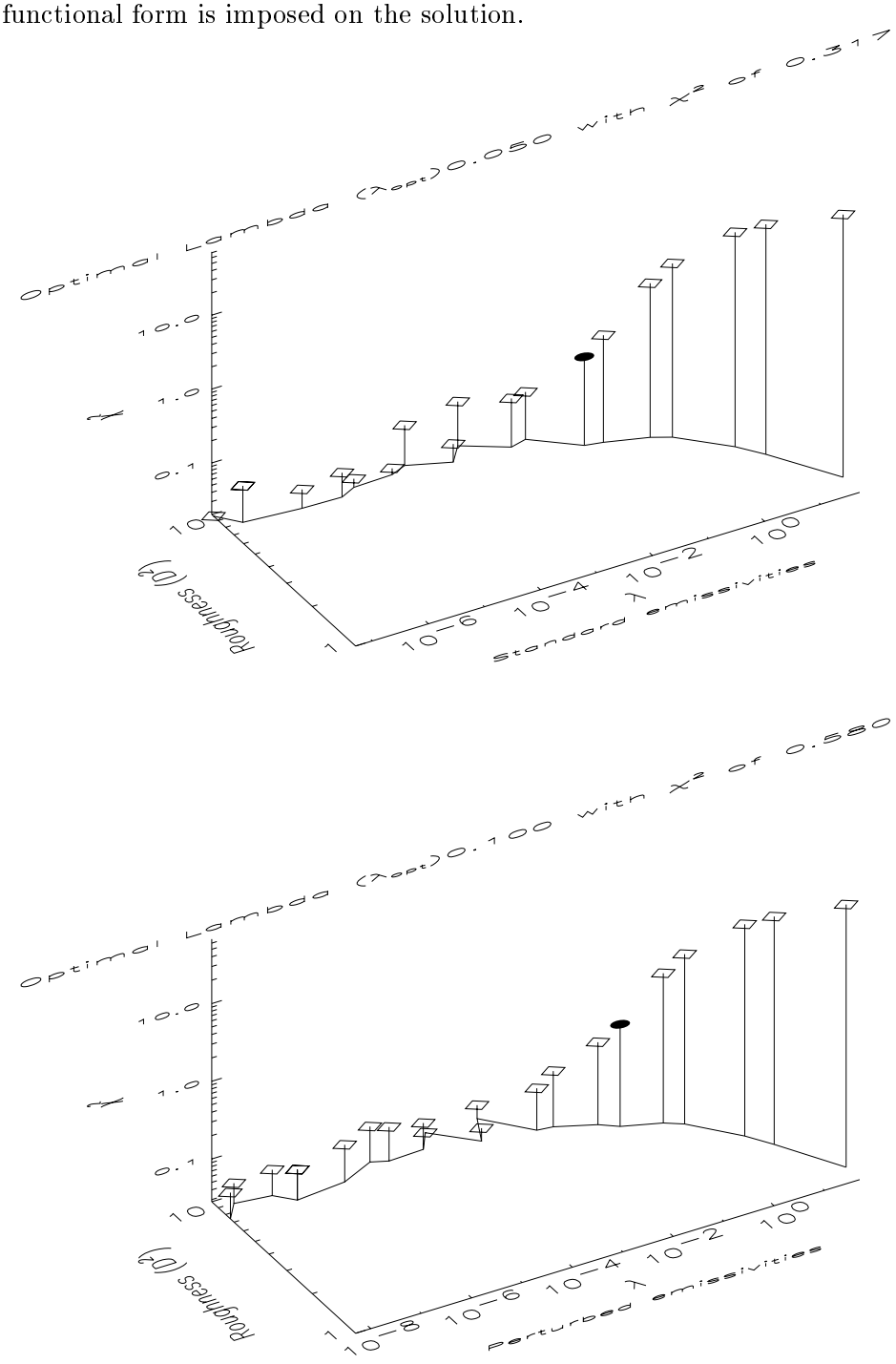

Figure 4.19: The global results of the RIT test on model 2 of Section 4.3.1 using a **Maximum Entropy** smoothing functional are presented here with quantities as described in the corresponding figures above. In the upper plot, for standard emissivities, λ_{opt} has a value of 0.05 which has an associated χ^2 of 0.317. Likewise, the lower plot, for perturbed emissivities, λ_{opt} has a value of 0.10 which has an associated χ^2 of 0.580.

4.4 Application of the RIT to SERTS-89 data

Now we present details of the application of the RIT to data acquired by the aforementioned SERTS mission flown on May 5^{th} 1989 (hereafter SERTS-89). The aim of this analysis being the recovery, and comparison, of the differential emission measure in T_e for the transition region and corona ($5 \leq \log_{10} T_e \text{ K} \leq 7$) with those published, using the *same* data previously (Brickhouse et al. 1995; Landi & Landini 1997; Lanzafame et al. 1998).

The observations made during the SERTS-89 flight concentrated on one active region (NOAA AR5464) from which a total of 269 emission lines were measured[7] in a wavelength range covering 170-450 Å. Details of the data reduction and calibration can be found in Thomas & Neupert (1994).

Application of the RIT to this particularly interesting dataset may resolve discrepencies in the DEM analysis presented in each of the papers given immediately above. Each author published recovered forms for $\xi(T_e)$ functions showing different functional structure (having direct, potentially incorrect, implications for the physical structure of the plasma itself)

[7]The absolute intensities of the lines were averaged over the entire duration of the flight.

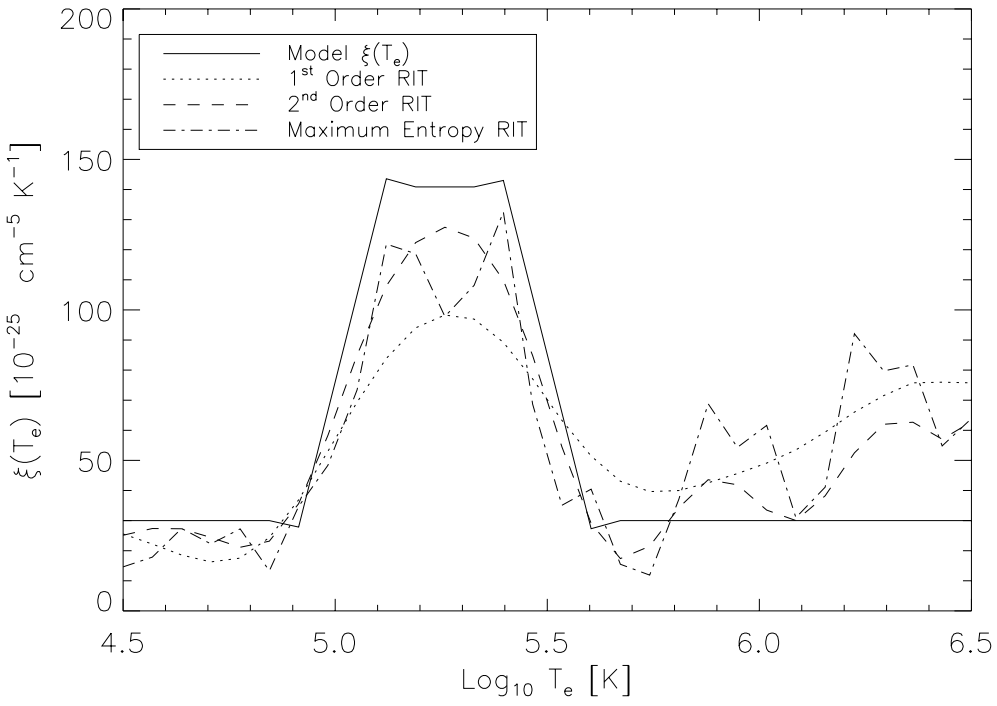

Figure 4.20: This plot shows the specific results of the Maximum Entropy smoothing functional of equation (4.42). It is clear that the RIT is adaptable to different forms of $\Phi(f(s_e))$.

between $6 < \log_{10} T_e < 7$. The DEM analysis presented by Brickhouse et al. (1995) found that $\xi(T_e)$ had a *single* temperature peak whereas the analysis of Landi & Landini 1997 (see top of figure 4.21) was *triple* peaked over the same region. As an aside, the DEM analysis of Brosius et al. (1996) for SERTS-91 and SERTS-93 averaged active region spectra suggested that $\xi(T_e)$ should be *double* peaked over this temperature range. The analysis presented in Lanzafame et al. (1998) (also advocating the *single* peak $\xi(T_e)$, see bottom of figure 4.21) demonstrated that spurious results such as the above may arise from the same theoretical uncertainties in the line emissivities discussed by Judge et al. (1997) but also from the use of an integral inversion technique with an arbitrary smoothing functional. Here, we see what the RIT can recover from the same data.

As far as our analysis is concerned, we concentrate on the functional form of $\xi(T_e)$ recovered by the RIT (using all three smoothing functionals) over a fixed number of generations (10,000) for a 30 point temperature discretisation. The line emissivities are calculated at a fixed electron density ($n_e = 5 \times 10^9$ cm^{-3}; Lanzafame et al. 1998) obtained using a single density sensitive line ratio[8].

From the 'stronger' of the 269 emission lines observed we have selected 24 'Correlated' pairs of lines from the same ionisation stage (to minimise likely systematic uncertainties, leaving essentially **b-b** rate errors only). As above, we calculate the value of $\sigma_{l_{th}}$ for each ratio pair l using the recipe of Section 4.2.1 and present the details (wavelength, intensities, observational errors and fractional theoretical errors ϵ_l) of each ratio pair in Table 4.5. Note that all the values of ϵ_l lie in the 2-7 % range.

Figures 4.22 through 4.24 show the details of the RIT runs for the first, second order and ME smoothing functionals respectively. As for figure 4.4 we present these global results for the vector $\mathbf{v}$ $(= (\lambda,\ D^2,\ \chi^2))$ and indicate the solution with the minimum $\|\mathbf{v}\|_2$ on the curve by $\bullet$. This 'optimal' solution is plotted in the lower left of each figure. Similarly, in the lower right of the figures, we plot the calculated line ratios R_{calc} (indicated by $\diamond$) returned at the end of that RIT run against the observed line ratios R_{obs} (indicated by $*$) and their observational errors. These figures show the same optimisation trends (i.e. the minimum $\|\mathbf{v}\|_2$ coinciding with the best solution) as those presented in the previous section so we therefore adopt these solutions as the optimal forms of $\xi(T_e)$ on application of the RIT to the SERTS-89 data.

It is clear from figure 4.25 that the $\xi(T_e)$ functions recovered by all three smoothing

[8] Although we have discussed the use (and ambiguities) of line ratios as a single density measure of a clearly inhomogeneous plasma, for this discussion, we will go on 'blind faith'.

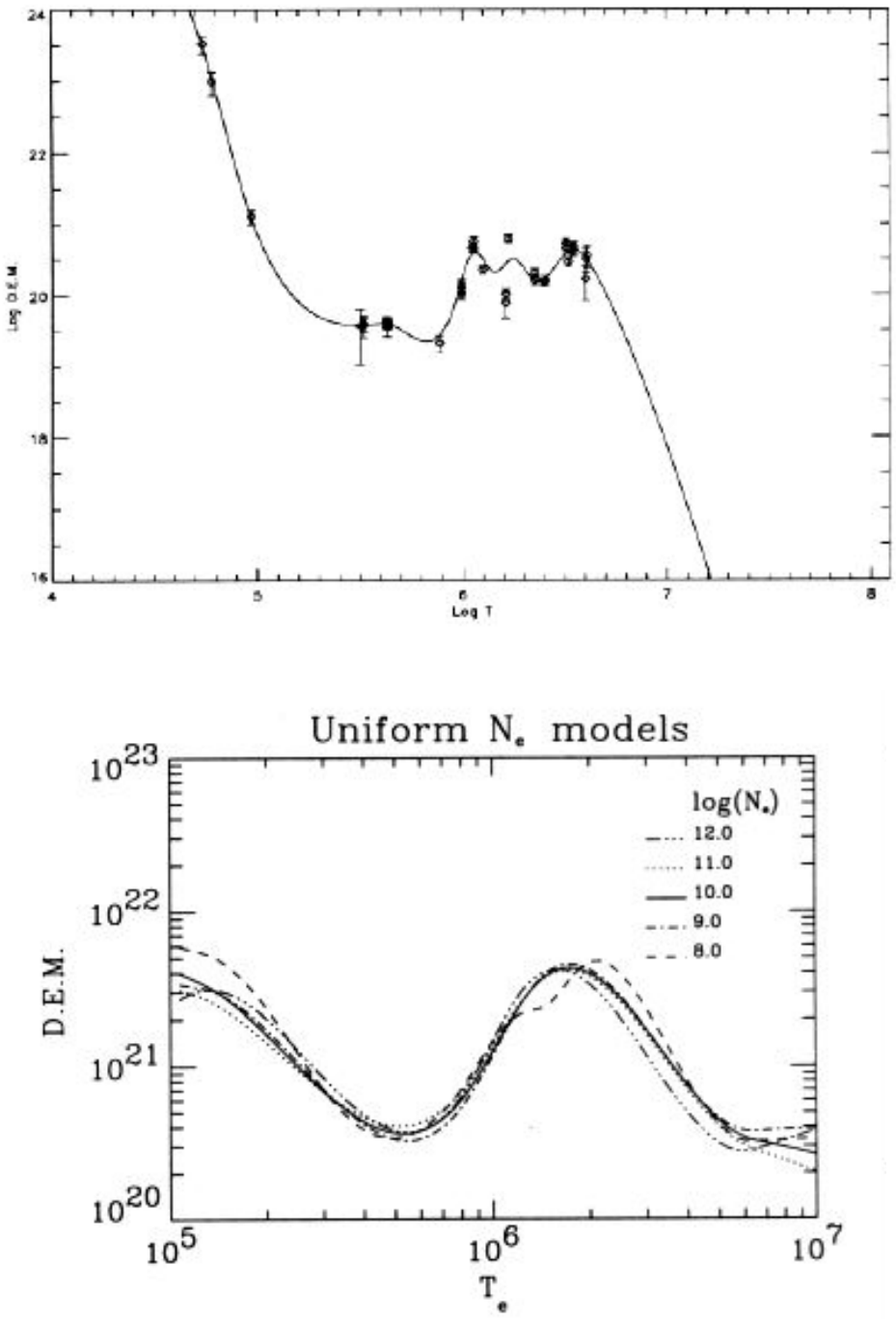

Figure 4.21: Differential Emission Measure functions for an averaged solar active region using EUV line intensities observed by SERTS-89. Top: (from Landi & Landini 1997) The DEM function recovered from the inversion shows a triple peaked form between $6 < \log_{10} T_e < 7$ calculated for $\log_{10} n_e = 10$. This is in contrast to that derived from the *same* data (bottom) by Lanzafame et al. (1998) which shows only a single peak in the same temperature range. The lower plot shows DEMs recovered for line emissivities calculated for a series of uniform electron densities to see if a similar multiple peaked function could be achieved.

Table 4.5: Details of the line pairs used in the RIT runs on $\xi(T_e)$ for EUV line emission data observed by SERTS-89. For each ratio pair we give the ion to which it belongs, the wavelengths of the lines used (λ Å), their intensities I^{obs} (units erg cm^{-2} sr^{-1} s^{-1}) the observed line ratio R_l^{obs}, the observational error σ_l^{obs} and a measure of the theoretical uncertainty in the line ratio ϵ_l (cf. Tables 4.1 and 4.3). The full line list for the SERTS-89 flight, from which this is extracted, can be found in Thomas & Neupert (1994).

#	Ion	λ_N Å	I_N^{obs}	λ_D Å	I_D^{obs}	R_l^{obs}	$\sigma_{l_{obs}}$	$\epsilon_{l_{th}}$
1	O III	374.075	14.4	374.164	4.9	2.939	1.313	0.062
2	C IV	384.031	8.6	419.713	12.4	0.694	0.301	0.023
3	O V	215.245	79.4	248.460	59.7	1.330	0.750	0.058
4	Ne VI	399.826	14.9	401.928	84.6	0.176	0.039	0.020
5	Ne VI	433.172	7.5	435.641	9.8	0.765	0.355	0.054
6	Mg VII	429.132	10.9	431.288	17.6	0.619	0.186	0.031
7	Mg VII	278.393	114.0	319.018	76.4	1.492	0.380	0.059
8	Mg VIII	315.015	253.0	338.983	53.8	4.703	0.933	0.043
9	Al IX	300.560	30.6	305.055	17.3	1.769	1.031	0.059
10	Al IX	384.950	7.0	392.425	15.3	0.458	0.154	0.021
11	Mg IX	368.057	1070.0	443.967	19.6	54.592	11.101	0.048
12	Si IX	290.687	33.2	296.113	208.0	0.160	0.077	0.043
13	Si IX	341.950	29.4	345.120	70.9	0.415	0.090	0.037
14	Si X	253.787	207.0	272.005	131.0	1.580	0.517	0.062
15	Si X	292.170	43.7	347.408	210.0	0.208	0.078	0.045
16	Si XI	303.326	2930.0	365.429	39.8	73.618	13.705	0.055
17	Si XI	361.410	23.7	371.492	14.5	1.634	0.517	0.049
18	S XII	288.420	135.0	299.540	47.2	2.860	1.195	0.034
19	S XIV	417.645	184.0	445.673	65.5	2.809	0.460	0.035
20	Ar XVI	353.860	7.7	389.069	12.8	0.602	0.307	0.056
21	Fe XVI	262.967	654.0	265.018	26.1	25.057	12.501	0.031
22	Fe XVI	335.401	10400.0	360.754	4320.0	2.407	0.542	0.051
23	Ca XVIII	302.205	25.3	344.760	13.6	1.860	0.890	0.060
24	Ni XVIII	291.970	357.0	320.537	152.0	2.349	0.411	0.027

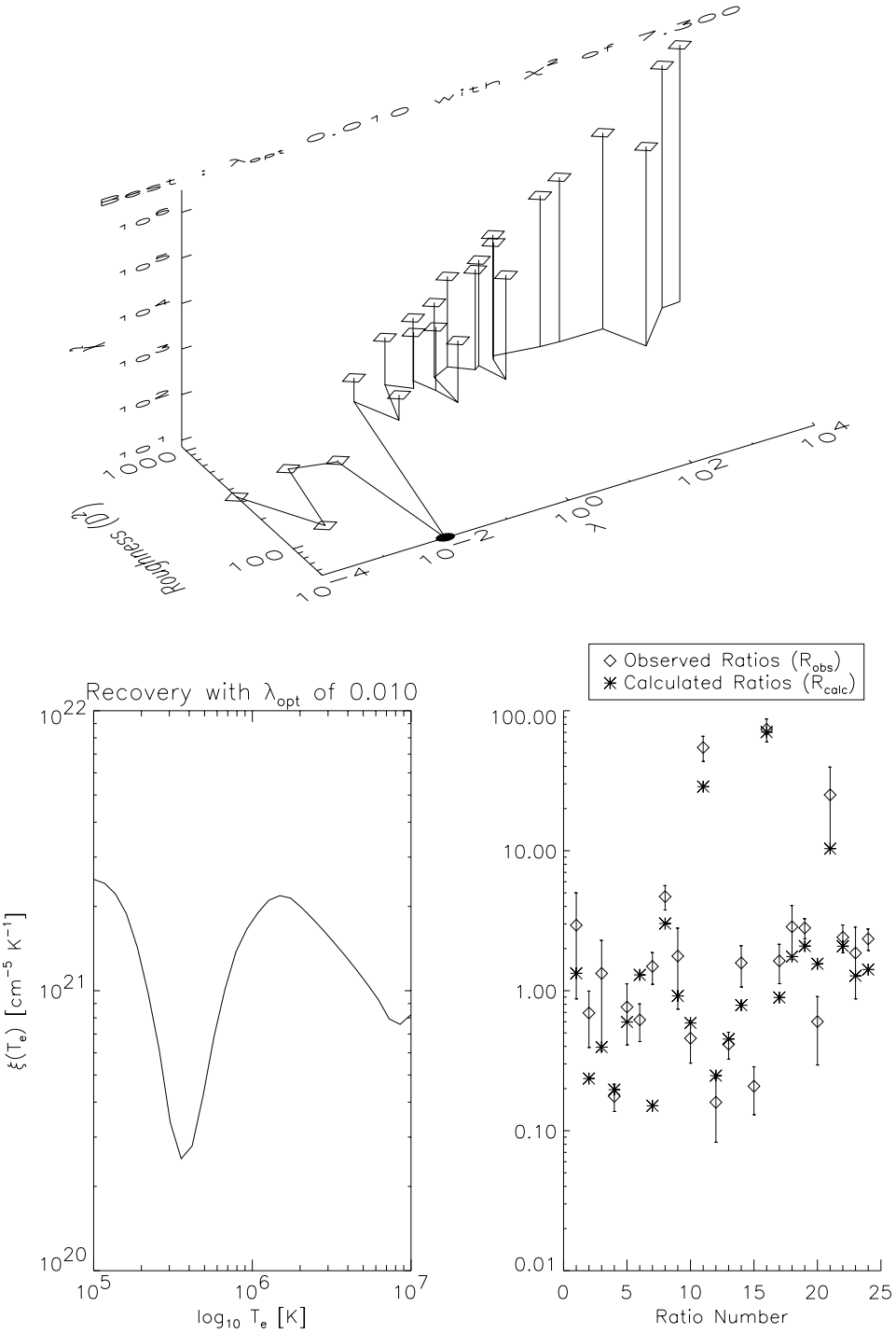

Figure 4.22: The global results of the RIT operating on SERTS-89 data with a **first order** smoothing functional. The upper plot (cf. figure 4.4) indicates (•) that the solution minimising vector $\mathbf{v} = (\lambda, D^2, \chi^2)$ is obtained for a smoothing parameter λ_{opt} of 0.010 and with an associated χ^2 of 7.300. The lower plots shows this optimal solution (left) and the recovery of the observed line ratios (R_{obs} and R_{calc} are given by $*$ and $\diamond$ respectively) with their associated observational errors (right). The solution shown clearly agrees with the single peak DEM in the $6 < \log_{10} T_e < 7$ region.

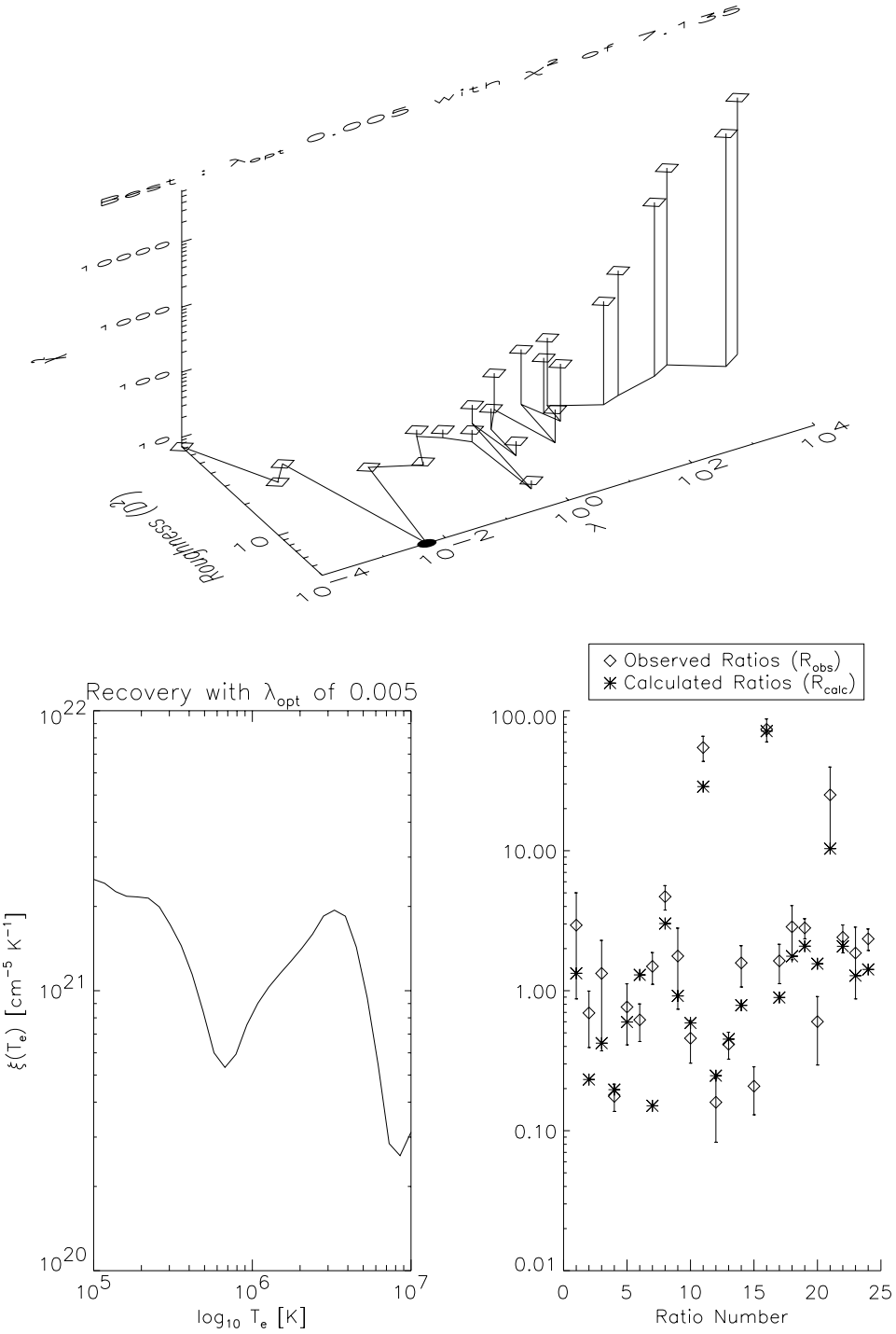

Figure 4.23: The global results of the RIT operating on SERTS-89 data with a **second order** smoothing functional. The upper plot indicates that the solution minimising vector $\mathbf{v} = (\lambda, D^2, \chi^2)$ is obtained for a smoothing parameter λ_{opt} of 0.005 and with an associated χ^2 of 7.135. The lower plots shows this optimal solution (left) and the recovery of the observed line ratios (R_{obs} and R_{calc} are given by $*$ and $\diamond$ respectively) with their associated observational errors (right). The solution shown clearly agrees with the single peak DEM in the $6 < \log_{10} T_e < 7$ region.

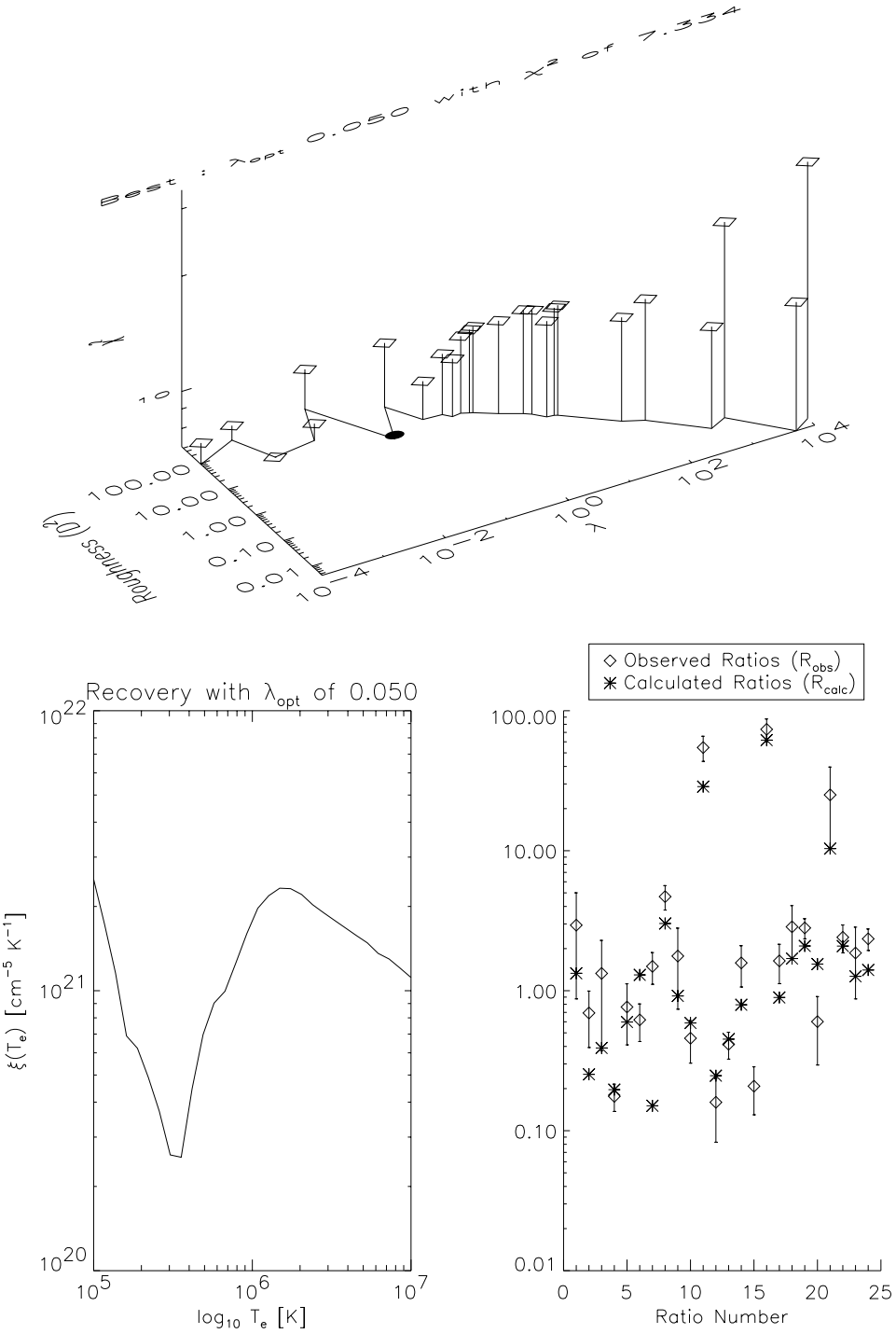

Figure 4.24: The global results of the RIT operating on SERTS-89 data with a **Maximum Entropy** smoothing functional. The upper plot indicates that the solution minimising vector $\mathbf{v} = (\lambda, D^2, \chi^2)$ is obtained for a smoothing parameter λ_{opt} of 0.050 and with an associated χ^2 of 7.334. The lower plots shows this optimal solution (left) and the recovery of the observed line ratios (R_{obs} and R_{calc} are given by $*$ and $\diamond$ respectively) with their associated observational errors (right). The solution shown clearly agrees with the single peak DEM in the $6 < \log_{10} T_e < 7$ region.

functionals clearly agree with the single peak DEMs of Brickhouse et al. (1995) and Lanzafame et al. (1998) and there is *no* clear evidence to support the triple peak model. Although this, again, highlights the severely ill-posed nature of the DEM inverse problem.

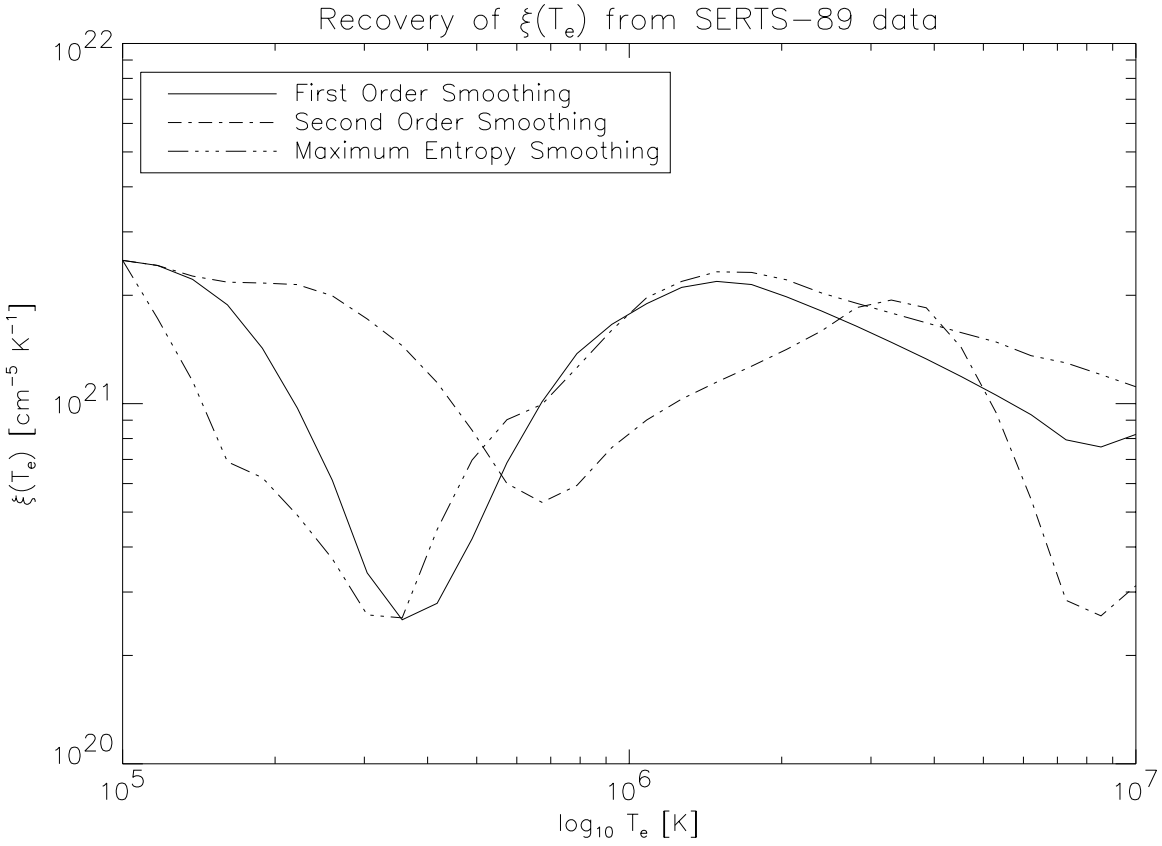

Figure 4.25: Plots of the three RIT solutions presented in figures 4.22, 4.23 and 4.24 for the three smoothing functionals. The solid line is the RIT solution for the first order smooothing functional with the dot-dash line and dot-dot-dash lines those solutions from the second order and Maximum Entropy smoothing functionals respectively. All of the solutions presented agree with the single peak DEM in the $6 < \log_{10} T_e < 7$ region. It is noted that there are considerable differences in the recovered functions (particularly first and second order) and to assess wether one solution is "better" than the others requires that we consider their final χ^2 values for the optimal smoothing parameter used. In this case, there is little difference in these final values and we can state that these solutions are statistically equivalent.

4.5 Discussion

We have shown that for an optically thin plasma, there is indeed a unique mathematical relationship between the 'mean' spectroscopic quantities $\langle n_e \rangle$, $\langle T_e \rangle$ and the differential emis-

sion measure functions ($\xi(T_e)$, $\zeta(n_e)$, and $\mu(n_e, T_e)$). These relationships are true provided certain assumptions hold regarding the nature of the emitting plasma, and the characteristic behaviour of particular line ratios, and show the equivalence between the full inversion and mean value methods.

For an optically thin, homogeneous, plasma and given a set of observed resonance line intensities, we have derived an expression that relates the 'mean' spectroscopic temperatures $\{\langle T_e \rangle\}$ and the discretised differential emission measure in temperature $\underline{\xi}$ (see equation (4.16)). Following a similar method we have obtained, for an isothermal plasma, an expression relating the 'mean' spectroscopic densities $\{\langle n_e \rangle\}$ and the discretised differential emission measure in electron density, $\underline{\zeta}$, which is given in equation (4.20). In the treatment of the general bivariate DEM function $\mu(n_e, T_e)$, Section 4.1.3 shows that we can obtain a representation of $\mu(n_e, T_e)$ when the conditions for Sections 4.1.1 and 4.1.2 occur simultaneously (*i.e.* situations where 'mean' densities and temperatures are actually defined). Essentially this means that for a large enough set of observed lines with different temperature and density characteristics, a relationship of the form of equation (4.27) will hold for the particular set of inferred 'mean' $\{\langle T_e \rangle, \langle n_e \rangle\}$ pairs.

Finally, we have discussed a potentially important inversion scheme, the Ratio Inversion Technique (RIT), based upon minimising differences between observed and computed *ratios*, instead of the more usual *intensities*. The RIT offers the possibility of removing large, systematic errors that may arise from uncertain ionisation balance, as have been suggested to explain solar data (Judge et al. 1995), and have been demonstrated to be the dominant source of error in standard inversions of line intensities (Judge et al. 1997). We can then clearly see that a method such as the RIT[9] is essential if the recovered DEM functions are to be used to further interpret solar UV line emission spectra from the SOHO or future missions.

[9] A version of the RIT code, written in Fortran-77, that used the aforementioned PIKAIA GA is given in Appendix A.2.

Chapter 5

Re-conditioning DEM inverse problems

This Chapter

In an inverse problem of any kind poor conditioning of the inverse operator decreases the numerical stability of any non-regularised solution in the presence of data noise. This chapter will show that, using a heuristic approach, we can improve the conditioning of the differential emission measure (DEM) inverse problems considerably by judicious choice of the integral operator and that there is indeed a set of solar UV/EUV emission lines that drastically increases the chance of obtaining unique electron density and temperature distributions of the emitting region of the solar atmosphere. This is essentially a way of formalising the choice of emission lines, and making a reproducible set of choices, instead of making the subjective line choices made in earlier work.

The material presented in Chapters 2 and 4 has shown that the choice of emission lines is critical to form accurate diagnostics of the solar plasma. The particular characteristics of different emission lines (or ratios thereof) will yield different information about the emitting plasma volume. Indeed, since the dawn of space-borne solar UV/EUV spectroscopy the emission lines observed display a qualitative physical dependence on the feature being observed. Today we want to obtain reliable[1] diagnostics of the outer solar atmosphere. Such diagnostics take the form of *distributions* of plasma characteristics (n_e, T_e, etc) such as the

[1] Reliable in the sense that we require the solution to be as unambiguous as possible; if there is a feature in the recovered diagnostic distribution (e.g, gradient, curvature or discontinuity) we require that this is not an artifact of the numerical processing.

differential emission measure (DEM) functions discussed in the previous chapters. However, the process of inferring such a distribution of quantities is not one to be taken lightly since it is 'booby-trapped' with numerical instability and non-uniqueness. Previous work on these DEM problems have concentrated on the physical nature of the emission lines used, including work discussed in this thesis. However, we present a new approach in an effort to reduce, as much as possible, the ambiguity of such poorly conditioned inverse problems. We do this by considering the mathematical properties of the solar UV/EUV emission lines and not only their physical properties, i.e. the n_e, T_e sensitivity of each.

We have seen that, as far as inverse problems are concerned, the uniqueness and numerical stability of the solution is acutely sensitive to the conditioning of the resulting matrix equation. For the DEM inverse problems we must maximize the 'potential' of the data inversion and we will see that the choice of lines[2] to analyse will help achieve this goal. The freedom present in the DEM problems (construction of kernel matrices) can be exploited to disclose an optimal set of lines from the UV/EUV lines in the wavelength range of the SOHO CDS/SUMER instruments ($150 - 1610$ Å).

For example, consider the emission line labelled l with total integrated line intensity (I_l) given by the double integral in terms of n_e and T_e as

$$I_l = \int_{T_e} \int_{n_e} K_l(n_e, T_e)\, \mu(n_e, T_e)\, dn_e\, dT_e. \tag{5.1}$$

where $\mu(n_e, T_e)$ and $K_l(n_e, T_e)$ are as defined previously. Equation (5.1) can, as demonstrated in Chapters 2 and 4, be reduced to a univariate Fredholm integral equation of the form

$$I_l = \int_{s_e} K_l(s_e)\, f(s_e)\, ds_e \tag{5.2}$$

where (for plasma characteristic $s_e = n_e, T_e$) $K_l(s_e)$ is the line emissivity[3] and $f(s_e)$ is representative of the differential emission measure in either n_e or T_e. We have noted previously (Chapter 2) that one such equation will only allow us to reliably specify $f(s_e)$ at one chosen s_e point since we only have one data point I_l. So, if we observe N emission lines and discretise equation (5.2) at M points in s_e we will have to built a matrix equation around an $M \times N$ matrix ($K_{M\times N}$), the *kernel* matrix[4] of the integral equation.

[2]For brevity, the term 'line' will be used for emission line.

[3]The term 'emissivity' is used throughout this chapter to represent the line emission coefficient normalised by n_e^2.

[4]For the rest of this chapter we will use the term 'kernel' to mean the kernel matrix of the integral equation, often by simply referring to K.

Once the kernel matrix has been 'constructed' we must assess the degree of numerical instability and non-uniqueness present in the discretised form of equation (5.2). For observational errors δI_l in the total integrated line intensity I_l and assuming that K is free of error, the fractional error δfin $f(s_e)$ is given by (cf. equation (2.27))

$$\frac{\|\delta f\|}{\|f\|} \leq C_K \cdot \frac{\|\delta I\|}{\|I\|} \tag{5.3}$$

where $\|\cdot\|$ is any Euclidean norm and C_K is the *condition number* of the kernel.

We have seen, in Chapter 2, that the perfectly conditioned matrix has $C_K = 1$ and is the identity matrix of order n ($I_{n\times n}$) or any diagonal matrix. Similarly, we have seen that kernel matrices with a *high* level of linear dependence of their rows have very large condition number which tends to ∞ as the degree of linear dependence increases. So, given the critical dependence of solutions $f(s_e)$ on C_K, we have two *hypothetical* questions to answer :

Q1 "Is the order in which discretised emissivities are placed in the kernel matrix important ?"

A1 No, the properties of matrices (determinant, condition number, etc) remain unchanged by *elementary row operations* such as row interchange (see, e.g., Whitelaw 1983).

Q2 "When we are constructing the kernel matrix from the number (X) of possible lines in the observed spectrum, which N ($N < X$; using each only once) should we choose such that the conditioning of the $f(s_e)$ inverse problem is optimised, for each case of s_e ? If such an optimal subset exists, what physically makes those lines better suited than those not chosen ?"

A2 These are precisely the questions this chapter will address.

So, adopting the scenario of Question 2, the problem has a potentially massive number ($\binom{X}{N} = \frac{X!}{N!(X-N)!}$) of possible "solutions". In this case a solution is a vector $\mathcal{V} = (v_1, v_2, \ldots, v_N)$ with each element, v_i, a unique line identifier (each v_i can only appear once in $\mathcal{V}$) such that the condition number C_K of K is minimised. Indeed, to understand what such solutions mean we require a mental picture of what the difference between a poorly conditioned kernel and a well conditioned one is. Recalling from above that the condition number is, if only philosophically, directly proportional to the degree of linear dependence in the rows of K. The conflicting requirements of K are clear, large numbers of lines ensures that the coverage of the s_e domain is good but there is a high value of C_K, put less lines

in the study and C_K decreases but the coverage is poorer. In constructing K we must find the happy medium. That is, we might expect the optimised kernel matrix K to have almost linearly independent rows; the ideal case being the delta functions, $\delta(x - x_0)$, of the identity matrix. However, for these inverse problems the line emissivities that form the rows of the kernel matrix have a finite amount of spread. We now discuss the form of this 'spread'.

To discuss the functional behaviour of certain emission lines we must return to equation (5.1) and recall from Section 2.2 that the line emissivity $K_l(n_e, T_e)$ (emission coefficient normalised to n_e^2) can be written as

$$K_l(n_e, T_e) = \frac{h\nu_l A_l}{4\pi} \frac{n_{u(l)}}{n_{ion} n_e} \frac{n_{ion}}{n_{el}} \frac{n_{el}}{n_H} \frac{n_H}{n_e} \quad \text{erg cm}^3 \text{ sr}^{-1} \text{ s}^{-1} \tag{5.4}$$

where A_l is the Einstein-A coefficient, $n_{u(l)}$ is the population density of the upper level of the transition, $\frac{n_{ion}}{n_{el}}$, $\frac{n_{el}}{n_H}$ and $\frac{n_H}{n_e}$ are the ionic abundance (ionisation fraction), elemental abundance, and relative abundance of H to electrons (taken to be constant) respectively. This relationship shows that the mechanism for upper level population will determine how $K_l(n_e, T_e)$ will behave as a function of n_e and T_e. We recall the simple 3-level atom, with level 3 metastable, of Section 2.2 as an aid to this description. Equations (2.78) and (2.79) give (upon solving for the non-LTE statistical equilibrium) the population densities of levels 2 (n_2) and 3 (n_3) in terms of the population density of the ground level (n_1). For the *resonance* line (transition from level 2 to the level 1) we have, assuming the population of level 3 to be negligable,

$$n_2 = \frac{n_e n_1 C_{12}}{A_{21}} \tag{5.5}$$

giving an emissivity ($K_{res}(n_e, T_e)$) of the form

$$K_{res}(n_e, T_e) = \frac{h\nu_{12} C_{12}}{4\pi} \frac{n_1}{n_{ion}} \frac{n_{ion}}{n_{el}} \frac{n_{el}}{n_H} \frac{n_H}{n_e} \,. \tag{5.6}$$

An *intersystem* line (transition from level 3 to level 1), involving the population density of the metastable level 3,

$$n_3 = \frac{n_e n_1 C_{13}}{(A_{31} + n_e C_{23})} \tag{5.7}$$

will have an emissivity ($K_{int}(n_e, T_e)$) behaving as

$$K_{int}(n_e, T_e) = \frac{h\nu_{31}}{4\pi} \left(\frac{C_{13}}{1 + \frac{n_e C_{23}}{A_{31}}} \right) \frac{n_1}{n_{ion}} \frac{n_{ion}}{n_{el}} \frac{n_{el}}{n_H} \frac{n_H}{n_e} \,. \tag{5.8}$$

where $C_{ij} = \kappa \Upsilon_{ij}(T_e) T_e^{-1/2}$ $(j > i)$ is the collisional excitation coefficient (s^{-1}), $\Upsilon_{ij}(T_e)$ is the Maxwellian averaged collision strength and κ is a numerical constant. The functional

behaviour of all the line emissivities in this chapter can be categorised as belonging to one or the other of these two classes. In this non-LTE plasma regime the electrons are assumed to belong to a Maxwell-Boltzmann distribution and populate the ground level preferentially. Such simplifications mean that C_{ij} is treated strictly a function of T_e. Indeed, at this point we can categorically state that :

- The assumption of a Maxwellian electron distribution ensures that $K_l(n_e, T_e)$ will be approximately Gaussian in the T_e domain or, more exactly, peaked around the temperature of maximum formation of the ionic stage to which that transition belongs with a full width at half maximum of 0.3 in $\log_{10} T_e$ (see, e.g., Jordan 1969).

- All lines will emit irrespective of the electron density of the plasma. Therefore $K_l(n_e, T_e)$ will cover the entire $n_e(10^8$ - 10^{12} cm$^{-3})$ domain of the upper solar atmosphere, but their functional behaviour will depend critically on the transition from which they arise.

For the cases considered in this chapter we will consider only univariate emissivities, $K_l(s_e)$. The physical reasons directly above ensure that a finite amount of 'overlap', and hence linear dependence in the kernel matrices will occur; we will *never* obtain a DEM kernel matrix with $C_K \approx 1$. We might presume, at this point, that the 'best' kernel matrices have rows which, when summed, cover the s_e domain *uniformly*. Conversely, we would expect the 'poorest' kernel matrices, those with the highest condition numbers, to contain rows which, when summed, cover little of the s_e space and are highly non-uniform in appearance. We will return to the discussion of these properties in due course.

The emissivities used in this chapter (as in the previous one) belong to strong transitions in the wavelength range $150 - 1610$ Å for ions of various iso-electronic sequences from various atoms including : Carbon (II - IV), Iron (XII - XV), Magnesium (VI - X), Neon (VI - VIII), Nitrogen (II - V), Oxygen (II - VI) and Silicon (III - XII). The precise details of the iso-electronic transitions used are given in Table 5.1 and there are 133 lines in total.

We will perform this analysis by considering the variation of kernel condition number C_K (of Section 2.1) with different choices of lines. The calculations presented here were made using a variation on the GA heuristic search algorithm presented in Chapter 3 called SELECTOR. The power of this idea arises from the fact that, in the DEM inverse problems, we can arrange the kernel matrix K in *any* way we choose, provided that we have observed the relevant lines in the spectrum.

We discuss the basic mechanism of the SELECTOR GA in Section 5.1 with details of trials

Table 5.1: The details of the emission lines used to produce the emissivities in this chapter. Only lines in the range of the CDS and SUMER instruments on SOHO were used (150- – 1600 Å). Notable exceptions are the lines belonging to the iron ions (Fe XII-XV).

Sequence	Transitions	Ions
Lithium	$2s - 2p$, $2s - 3p$	C IV, N V, O VI, Ne VIII, Mg X, Si XII
Beryllium	$2s^2\ ^1S - 2s2p\ ^3P,\ ^1P$	C III, N IV, O V, Ne VII, Mg IX, Si XI
	$2s2p\ ^3P,\ ^1P - 2p^2\ ^3P$	
Boron	$2s^2 2p\ ^2P - 2s2p^2\ ^4P,\ ^2D$	C II, N III, O IV, Ne VI, Mg VIII, Si X
	$2s2p^2\ ^4P - 2p^3\ ^4S$	
Carbon	$2s^2 2p^2\ ^3P - 2p^3\ ^5S,\ ^3D$	O III, Mg VII, Si IX
	$2s^2 2p^2\ ^3P - 2s^2 2p^2\ ^1D,\ ^1S$	
Nitrogen	$2p^3\ ^4S - 2p^3\ ^2D,\ ^2P$	Mg VI
Sodium	$3s - 3p$	Si IV
Magnesium	$3s^2\ ^1S - 3s3p\ ^3P,\ ^1P$	Si III
	$3s3p^2\ ^3P - 3p^2\ ^3P$	

on the two 'diagnostic' DEM inverse problems in Sections 5.2 and 5.3. These trials show that there are indeed subsets of the X observable lines that make the DEM inverse problems considerably better conditioned than using *all* of the observable lines in the inversion (for a fixed number of solution points M). Indeed, for each of the test cases, we will discuss the properties of the emission lines that make them better than others in the inverse problem framework.

5.1 Specifics of SELECTOR

We present a Genetic Algorithm (GA) method SELECTOR[5] that, unlike those discussed in the previous chapters does not minimise with respect to a standard χ^2 measure but instead minimises C_K, the condition number of the DEM inverse problem kernel matrix. Therefore, our simple algorithm will choose a subset of N lines from the 133 lines present in the HAO-diaper calculations[6] such that C_K is minimised.

We note that this (combinatorial) condition number minimisation problem involves the identification of a subset of N distinct elements from a search list of X ($> N$) possible choices where the ordering of these elements is *not* important. A GA naturally lends itself to the optimisation of such a problem but becomes a more powerful tool when analogy is drawn (algorithmically and computationally) between this problem and that of the *Travelling Salesman Problem* (TSP) discussed in Chapter 10 of Michalewicz (1994). Consider one possible statement of the TSP (given a set of N elements, the TSP requires to find the permutation of those elements that minimises some criterion).

> "Before going on a road trip a salesman will plan his journey to be: cost effective, of minimum duration and yet to bring in as much business as possible. If he is required to visit X towns, in no specific order, passing through each only *once* to make his calls. Which route should he take ?"

It would therefore seem obvious to take advantage of the methodology used for the TSP to construct the list of distinct elements from which we hope to construct the DEM kernel matrix with the smallest possible condition number.

The construction of such a list of distinct elements from a reference list with possible

[5] The actual Fortran 77 code for the SELECTOR GA is included in Appendix B.2.

[6] It is noted that the *Chianti* database of Dere et al. (1997) would also provide appropriate data for these calculations.

non-distinct entries is readily carried out using the **Ordinal Representation** scheme of Michalewicz (1994). The task is to extract N (fixed) distinct elements from a list **S** of X possible values, where $S_m = m,\ j = 1, ..., X$. An *ordinal vector* **e** is made up of N elements (e_n) with values in the range $1 \leq e_n \leq X - n + 1$. The corresponding *element vector* **E** is constructed according to the following iterative procedure :

do $n = 1, N$

$E_n := S_{e_n}$ $\star$

$S_k := S_{k+1}, \quad k = n, ..., X - n - 1$ $\star\star$

enddo

$(\star\star)$ has the consequence that S_{e_n} is removed from the list, and the list size is reduced by one at each iteration. Consider for example a situation where $N = 10$ distinct elements must be extracted from a reference list of $X = 40$ possible values. The ordinal vector

$$\mathbf{e} = (4, 5, 1, 29, 26, 11, 31, 8, 22, 5)$$

decodes into

$$\mathbf{E} = (4, 6, 1, 32, 29, 14, 37, 11, 27, 8)\ .$$

This ordinal representation scheme has some attractive characteristics. It is quite simple to implement, and having the e_j's uniformly distributed in their allowed bounds results in a uniform distribution of E_j's. However, one can easily verify that when pairs of **e** are acted upon by the one-point cross-over operator (see figure 3.1) , the e_j's located right of the splicing point can decode into E_j's not originally coded by the parent **e**'s. This is a direct consequence of the *leftward* shifting $(\star\star)$ associated with the encoding procedure. This is incompatible with the expected behavior of cross-over, which should lead to exchange of an intact subset of the E_j's. The only tolerable exception is when the cross-over operation introduce additional duplicate entries in the pair of **e** resulting from the cross-over operation. Likewise, under the ordinal representation uniform one-point mutation can potentially alter *all* elements of **E**. This makes the standard ordinal representation unsuitable for the present application.

A simple modification of standard ordinal representation, which we hereafter refer to as **Ranked Ordinal Representation** or simply as **ROR** (Charbonneau 1998 - Private Communication), can bypass the problems incurred using standard genetic operators. The

ROR method consists of *ranking* the ordinal vector $\mathbf{e}$ in decreasing order (so that $e_{n+1} \leq e_n$) prior to applying the ordinal algorithm ($\star\star$). So, under this scheme the ordinal vector

$$\mathbf{e} = (4, 5, 1, 29, 26, 11, 31, 8, 22, 5)$$

now decodes into

$$\mathbf{E} = (31, 29, 26, 22, 11, 8, 5, 6, 4, 1) \ .$$

Clearly, if $\mathbf{e}$ does not contain duplicate entries then $\mathbf{E} = \mathbf{e}$ (with $\mathbf{e}$ ranked). If however entry $e_n = e_{n+1}$ for some n ($< N$), then $E_n = e_n$ and $E_{n+1} = e_n + 1$ such that the symbolic algorithm given above becomes :

do $n = 1, N$

$E_n := \min(X - n + 1, S_{e_n})$

$S_k := S_{k+1}, \qquad k = n, ..., X - n - 1$

enddo

This actually introduces a slight bias toward high values of E_n, but for relatively small population sizes ($N_p \leq 100$, say) it remains statistically insignificant as compared to the realisation noise ($\propto \sqrt{N_p}$). Thus ROR will ensure, for small population sizes, that the line lists constructed throughout the fixed generation number run will have each selected line represented only once.

The algorithmic steps of the SELECTOR GA are :

1. Using the ROR technique, choose N lines randomly for each member of the population assuring that each line appears only once.

2. Calculate C_K for each member of the population using either

   (a) the condition number estimate of Cline et al. (1979) (see discussion in Appendix B.1 and Golub & Van Loan 1989).

   (b) a full singular value decomposition (SVD) of the matrix to calculate $C_K = \frac{\sigma_{max}}{\sigma_{min}}$.

3. Rank the population according the condition number.

4. Perform breeding in the population using the cross-over and mutation operators (see Chapter 3).

5. Check that the maximum number of generations has not been reached, then return to step 2, else proceed.

6. Return the set of UV/EUV lines that minimises the condition number of the kernel matrix.

5.2 Optimising the $\xi(T_e)$ inverse problem

The interpretation of UV/EUV emission spectra from solar and astrophysical plasmas often hinges on the inference of the emission measure differential in T_e, $\xi(T_e)$. Recalling the discussion of Section 2.2.1.1 we can simply define

$$\xi(T_e) = \int_{S_{T_e}} \frac{n_e^2}{|\nabla T_e|} \, dS_{T_e} \ , \tag{5.9}$$

where S_{T_e} is a surface of constant T_e within the emitting volume of plasma. The emission measure differential in temperature can be taken, literally, as the temperature gradient weighted mean square electron density.

We see that for a homogeneous plasma, with $n_e = n_o = 10^9$ cm^{-3}, the double integral of equation (5.1) reduces to the single integral of equation (5.2) with $s_e = T_e$ and $K_l(n_o, T_e) = K_l(T_e)$ i.e.

$$I_l = \int_{T_e} K_l(T_e) \, \xi(T_e) dT_e \ . \tag{5.10}$$

The $\xi(T_e)$ function is the solution of this Fredholm integral equation of the first kind. Numerical errors in the emission line intensities (δI_l) of this inverse problem will, once discretised, induce errors ($\delta\xi$) in the solution ξ of a magnitude given by equation (5.3).

The majority of publications containing derivation of $\xi(T_e)$ functions from observed UV/EUV line intensities from the Sun or other stars adopt the "invert for all lines" or "all-lines" approach (see, e.g., Kashyap & Drake 1998; Lanzafame et al. 1998). This method involves the use *every* emission line observed to construct the kernel matrix (K) for the inverse problem and hence perform the numerical inversion and obtain $\xi(T_e)$. The *vast* majority of such publications completely neglect the effect of error propagation from data to solution because of poor conditioning of the inverse problem kernel matrix. To counter the apparent neglect of just how poorly conditioned this inverse problem is we will choose, using the GA approach discussed above, an optimal subset of emission lines. The ultimate aim being that this optimal set will have a significantly lower kernel condition number than that of the "all-lines" approach. To this end we consider the selection of the 30 emission lines[7] (from the 133

[7] The number of lines used in the calculation is arbitrary, but taken to be 30 for this discussion.

possible lines) that minimises the condition number C_K of the $\xi(T_e)$ inverse problem using a 30 point T_e discretisation. We show that this optimal subset does indeed have a significantly lower condition numbers than those using all of the observed lines.

So, what is the physical reasoning behind the statement that there is some subset of the 133 lines that have a significantly better conditioned kernel than other subsets? To answer this question we must look at the functional behaviour of the line emissivities as functions of T_e. For a resonance line in the simple 3-level atom, with n_e constant ($n_o = 10^9$ cm^{-3}), inspection of equation (5.6) will show that the functional dependence of $K_{res}(T_e)$ is determined by the interplay between the population of the ground level (itself dependent on the abundance of the ionisation stage to which the transition belongs) and the collisional excitation rate of the transition. The approximation of a Maxwellian-Boltzmann electron distribution will ensure that $K_{res}(T_e)$ is a peaked function of T_e with its maximum at some temperature T_o, the value of T_e where the ionic abundance is a maximum for this particular n_o.

As can be appreciated from equation (5.8) the T_e dependence of an intersystem line's emissivity is not quite as trivial. Equation (5.8) shows that the critical electron density n_{e_c} (where $n_e C_{23} \approx A_{31}$) plays an important role. The value of n_{e_c} is different for each transition. If we have for a particular intersystem transition the case where $n_o \ll n_{e_c}$ the temperature dependence of $K_{int}(T_e)$ will be determined solely by the numerator, and will resemble $K_{res}(T_e)$ and be a strongly peaked function. However, another intersystem transition may depend on a metastable level which has $n_{c_e} > n_o$ and then both the denominator and numerator must be considered as important terms. $K_{res}(T_e)$ can be approximated from the collision strengths (Υ_{13} and Υ_{23}) and ionisation balance of the relevant transition by

$$K_{int}(T_e) \approx \frac{\Upsilon_{13}(T_e)\, T_e^{-1/2}}{1 + \Upsilon_{23}(T_e)\, T_e^{-1/2}}\, \frac{n_{ion}}{n_{el}}. \tag{5.11}$$

The resulting function has a roughly Gaussian shape, peaked at the temperature of maximum ionic abundance, but skewed shortward of T_o. Figure 5.1 demonstrates these slight differences in functional dependence on T_e for a resonance line (765.147 Å) and intersystem line (1486.496 Å) of N IV.

Figure 5.3 shows the form of a typical run of SELECTOR for the $\xi(T_e)$ inverse problem over 2000 generations with 100 individuals in the population. We see the normalised linear superposition ($\widehat{S_j(T_e)}$) of all the selected emissivities $K_l^*(T_e)$, normalised to the maximum

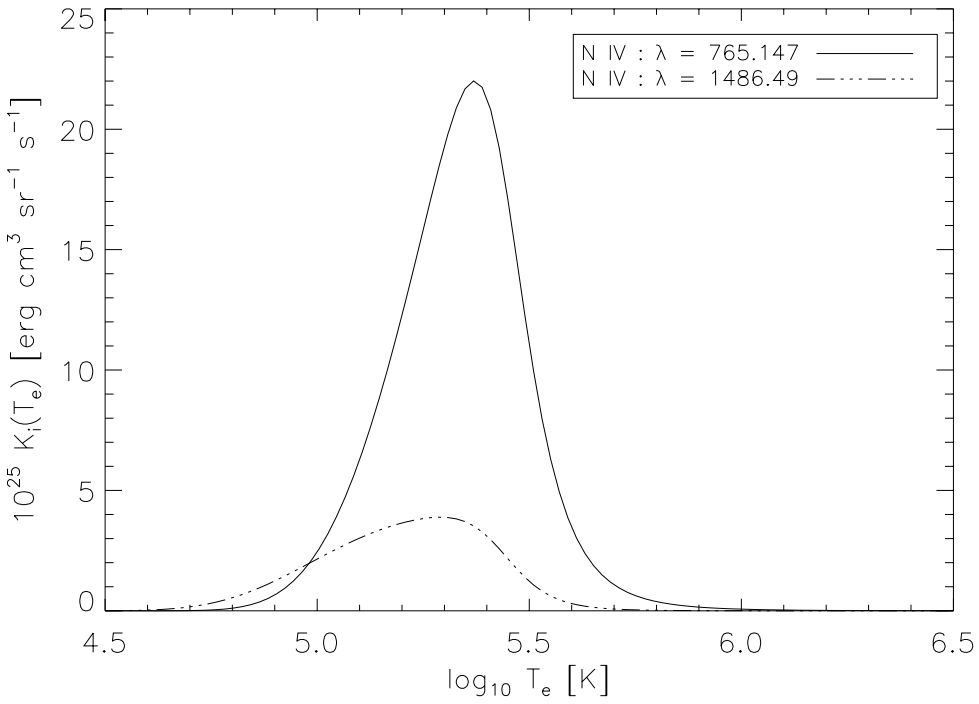

Figure 5.1: The emissivities of a resonance line (solid line) and intersystem line (dashed line) as functions of temperature only. These are for lines of N IV (wavelengths 765.147, 1486.496 Å) calculated for a electron density of $n_o = 10^9$ cm^{-3}.

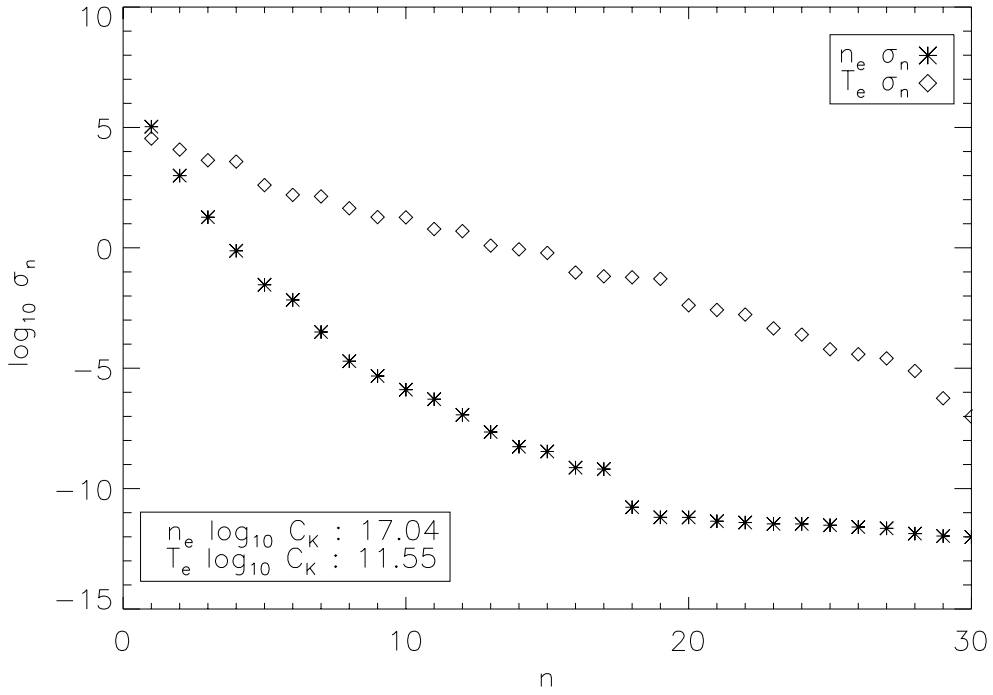

Figure 5.2: The singular value (σ_n) distribution of the matrix constructed from *all* 133 emissivities considered in this problem for both, $\zeta(n_e)$ and $\xi(T_e)$ inverse problems with a 30 point discretisation. Using the condition number estimate employed by SELECTOR we obtain values of $\log_{10} C_K = 18$ and 12 for density and temperature kernels respectively.

element $K_l^{Max}(T_e)$, at generation j is given by

$$\widehat{S_j(T_e)} = \frac{\sum_{l=1}^{M} K_l^*(T_e)}{Max(S_j(T_e))} \tag{5.12}$$

versus generation number. M is the number of points over which the emissivities are discretised ($M = 30$ in this case). For this sample run we see that the minimum ($\log_{10} C_K = 4.4972$) is very much smaller in comparison to the condition number ($\log_{10} C_K = 11.55$) of the "all--lines" approach (see figure 5.2). The set of emission lines chosen at the end of this *single* run of SELECTOR may not form the 'optimal' choice that minimises C_K, as we will see below, but obtaining that set (in a evolutionary sense) displays certain characteristics mentioned above. For example, consider figure 5.4 where we have plotted ($\widehat{S_j(T_e)}$) for generations $j = \{1, 500, 1000, 1500, 2000\}$. The upper portion of this figure exhibits a feature mentioned above about the nature of the conceptually well conditioned kernel matrix, *i.e.* the superposition of the rows, taken in projection, should span the domain as uniformly as possible. By comparing the upper and lower panels of figure 5.4 we can see how the percentage of coverage (PC_j)

$$PC_j = \frac{\int_{T_e} \widehat{S_j(T_e)}\, dT_e}{M} \tag{5.13}$$

varies with generation. The lower panel clearly shows that percentage of coverage is related to the condition number: greater uniformity of kernel coverage gives lower values of C_K.

A true test of this GA method for a problem of this combinatorial scale is to adopt a Monte Carlo approach[8]. This approach involves obtaining optimal sets of emission lines for many runs, each run having a different randomly chosen starting population (see Chapter 3 for more details) which is then encoded using the ROR technique to ensure that all the lines in the list are unique throughout the run.

Figure 5.5 shows that the Monte Carlo approach identifies lines that have particular properties, reducing the condition number of the kernel, and are chosen significantly more often than others. This figure, however, gives no clear indication that *any* of these lines occur together in the sets chosen, or in separate subsets, to form a kernel of significantly lower condition number. Identification of such a set is left to inspection of figure 5.6. Figure 5.6 shows the Monte Carlo runs vertically, each colour-coded[9] according with the value of C_K

[8] All Monte Carlo runs of SELECTOR were over 5000 generations to ensure that an optimal line set had been acquired (2000 of which are required, on average, to get within a factor of 2 of the optimal C_K).

[9] Colour coded using an colour table to make identification easier; white indicating the lowest condition number.

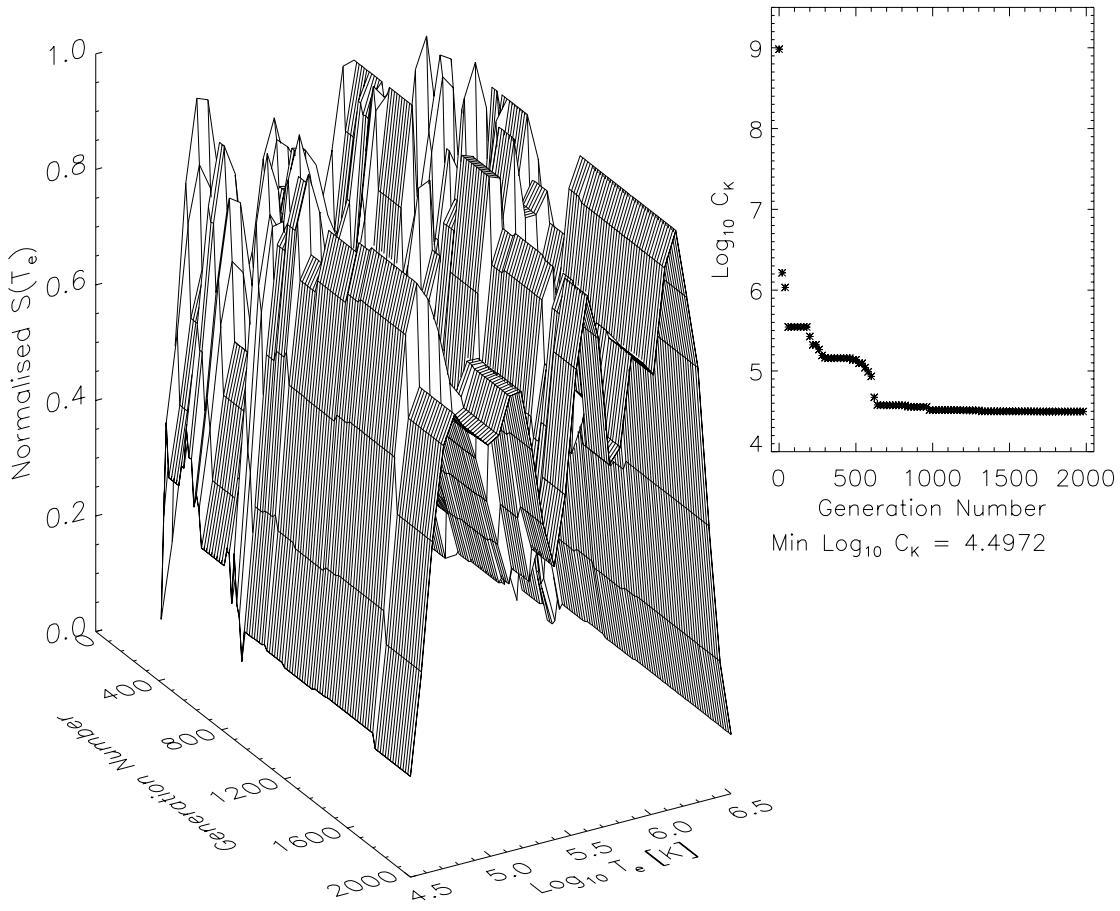

Figure 5.3: Representation of the evolution of the a sample solution with generation number. The calculation was performed over 2000 generations for a population of 100 individuals. For the 'fittest' individuals in the generation we plot the normalised $S_j(T_e)$; the linear superposition of kernels in that subset of the 133 lines, see equation (5.12) (*Each* kernel is normalised with respect to its maximum element and each $S_j(T_e)$ is then normalised to its maximum element such that an unbiased estimate of temperature coverage can be obtained). The inset of this figure (top right) shows the variation of C_K at each generation step.

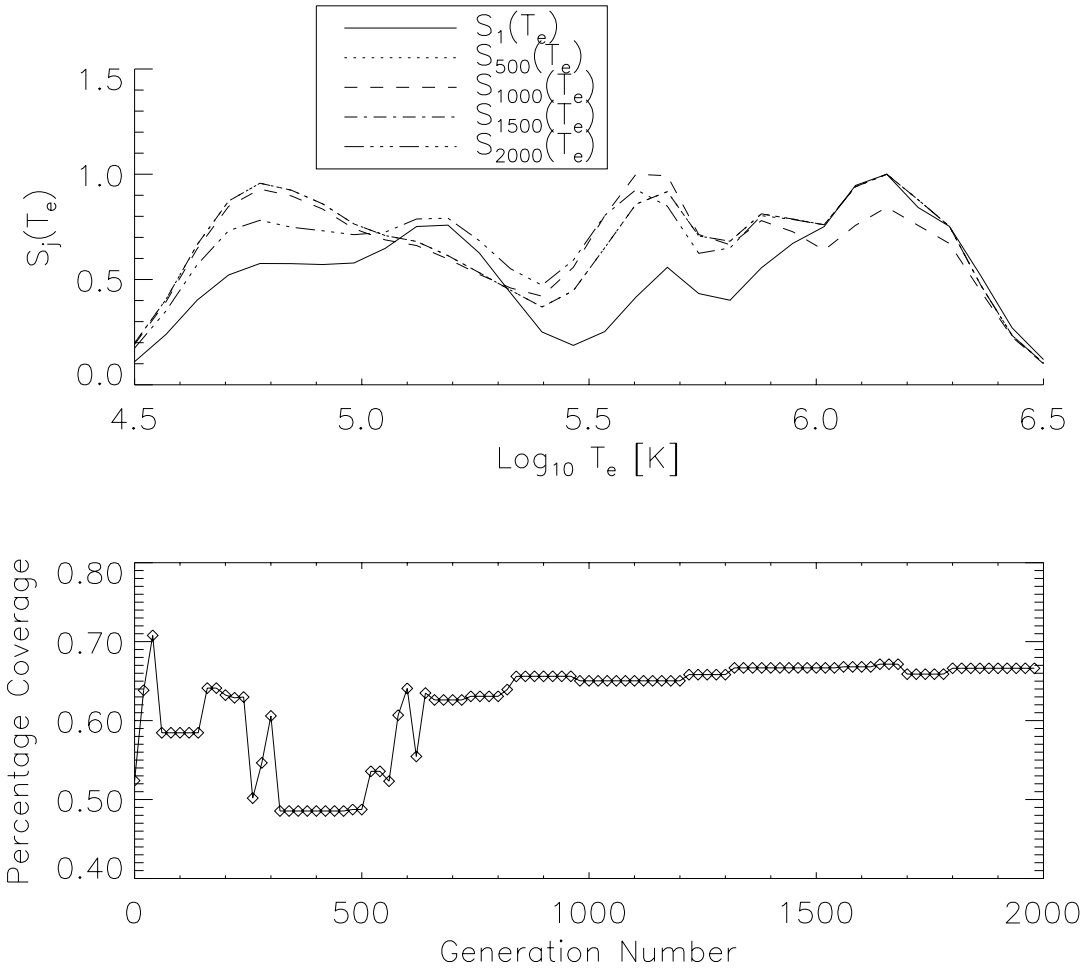

Figure 5.4: The top plot superimposes plots of $S_j(T_e)$ at points $j = \{1, 500, 1000, 1500, 2000\}$ during the evolutionary run shown above. The lower plot shows the coverage percentage of each $S_j(T_e)$ at every generation, this plot clearly demonstrates that better conditioned kernels have a more uniform 'amplitude' spread over the whole T_e domain.

belonging to the fittest individual at the end of the five thousandth generation. Of the 300, 5000 generation, runs of SELECTOR the value of C_K associated to the set of emission lines provided by run 106 is less than those of the other runs, with $\log_{10}\ C_K = 4.2709$. The data from figures 5.5 and 5.6 is collated in Table 5.2, with all lines of figure 5.5 above the mean selection frequency (72.1805) are given with the temperature at which $K_l(T_e)$ peaks (T_e^{max}) and if they belong to the set chosen in run number 106 then they are indicated by an asterisk ($*$).

The results of the Monte Carlo sequence of runs show that, for the $\xi(T_e)$ inverse problem, an optimal set exists and that the inverse problem will be considerably better conditioned than one using the "all-lines" approach. It is also demonstrated that the best kernels have the greatest degree of uniform coverage of the temperature domain (see, e.g, figure 5.7). This latter point ensures that DEM inversions performed using the optimised kernel will be independent of the regularisation method used (see Chapter 2). To clarify this statement we remember that regularisation smoothes discontinuous regions of the integration domain (i.e. it will try to fill in gaps and leaps with smooth polynomial functions). If the whole domain is uniformly sampled in the way we have discussed, regularisation will not be allowed to alias the recovered DEM function, $\xi(T_e)$.

Thus, to assess the validity of this GA analysis, we must perform an inversion for both the optimal subset of 30 (those identified in run 106) and the full set of 133 emission lines to compare the stability of the inverted solutions. These inversions are performed using a regularisation 'forward-backward' method. This method involves computation of line intensities (with appropriate errors, 15% in this test) for a given model $\xi(T_e)$ function. Then it is a simple case of employing a Tichonov regularisation algorithm (described in Chapter 2.1) with a range of smoothing parameters λ ($10^0 - 10^6$) to obtain a solution. Figure 5.8 clearly shows that the inversion performed with the optimal subset of lines is significantly more stable numerically than that obtained when using the "all-lines" approach, especially over the wide range of smoothing parameters used.

5.3 Optimising the $\zeta(n_e)$ inverse problem

Although not commonly sought after in astrophysical observations, the differential emission measure in electron density (cf. equation (5.9))

$$\zeta(n_e) = \int_{S_{n_e}} \frac{n_e^2}{|\nabla n_e|}\, dS_{n_e}\ , \tag{5.14}$$

Table 5.2: Details of the emission lines selected most at the end of the 300 Monte Carlo 5000 generation runs of SELECTOR. The emission lines included here are those with selection frequencies greater than the mean of 72.1805 counts. The lines indicated by an asterisk (∗) are those belonging to run 106, the set having the minimum value of $\log_{10} C_K = 4.2709$. Also given are the ions to which the line belongs, wavelengths λ (Å), the number of times the line was selected and the temperature at with the emissivity of the line peaks T_e^{max} (K).

Ion	λ (Å)	Count	$\log_{10} T_e^{max}$ (K)		Ion	λ (Å)	Count	$\log_{10} T_e^{max}$ (K)	
C III	977.020	77	4.8		C III	1175.98	91	4.8	
C III	1175.26	118	4.8		Mg VII	1189.82	76	5.7	
Mg VIII	352.460	111	5.8		Mg IX	443.403	136	5.9	
Mg IX	368.070	77	5.9		Mg X	609.793	108	6.0	∗
Ne VI	454.072	170	5.6		Ne VI	562.711	123	5.6	
Ne VI	1010.60	89	5.5		Ne VI	1006.09	136	5.5	
Ne VI	999.630	93	5.6		Ne VII	895.175	168	5.6	
Ne VII	562.993	144	5.6		Ne VII	887.279	119	5.6	∗
Si III	1206.49	154	4.7		Si III	1301.14	148	4.7	
Si III	1296.72	75	4.7		Si IX	344.951	74	6.0	∗
Si X	287.092	117	6.0	∗	Si XI	368.378	161	6.1	∗
Si XI	582.886	195	6.1	∗	Si XII	499.405	91	6.2	
N III	771.544	140	4.9		N III	991.502	140	4.8	∗
N III	771.900	74	4.9		N V	1238.82	122	5.1	
O II	539.085	81	4.6	∗	O II	540.012	142	4.6	
O III	833.715	80	4.9		O III	1666.14	115	4.8	
O IV	1397.23	100	5.1	∗	O IV	1401.15	108	5.1	
O IV	1407.38	73	5.1		O IV	624.618	127	5.1	
O IV	790.112	131	5.1		O IV	1399.78	107	5.1	∗
O IV	1404.80	114	5.1	∗	O V	761.128	128	5.3	
O V	759.441	126	5.3		O V	1213.80	102	5.3	∗
O V	1218.34	164	5.3		O VI	150.089	133	5.4	∗
Fe XII	1349.36	98	6.0		Fe XIII	1370.85	111	6.2	
Fe XIV	356.639	155	6.2		Fe XV	314.664	178	6.2	

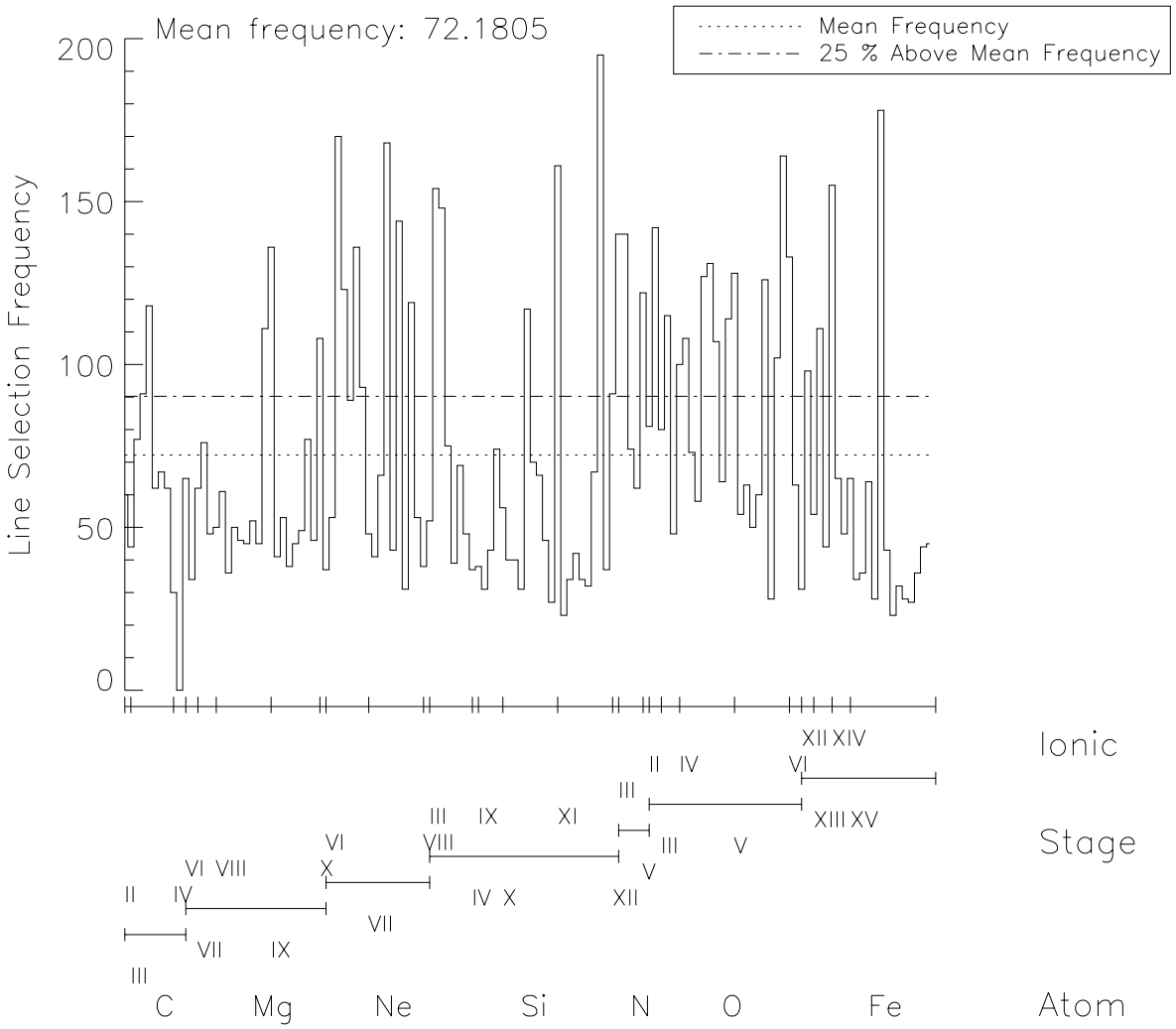

Figure 5.5: Monte Carlo histogram of line selection frequency versus line results for 300 runs of the SELECTOR. This clearly shows the existence of a subset of the 133 lines that have selection frequencies significantly greater than the mean. However, these do not form *the* optimal subset of 30 lines; taking the 30 most selected lines and computing C_K gives $\log_{10} C_K = 6.324$. The lower axis identifies the atom (large division) and ionisation stage (corresponding to the label) to which each frequency belongs.

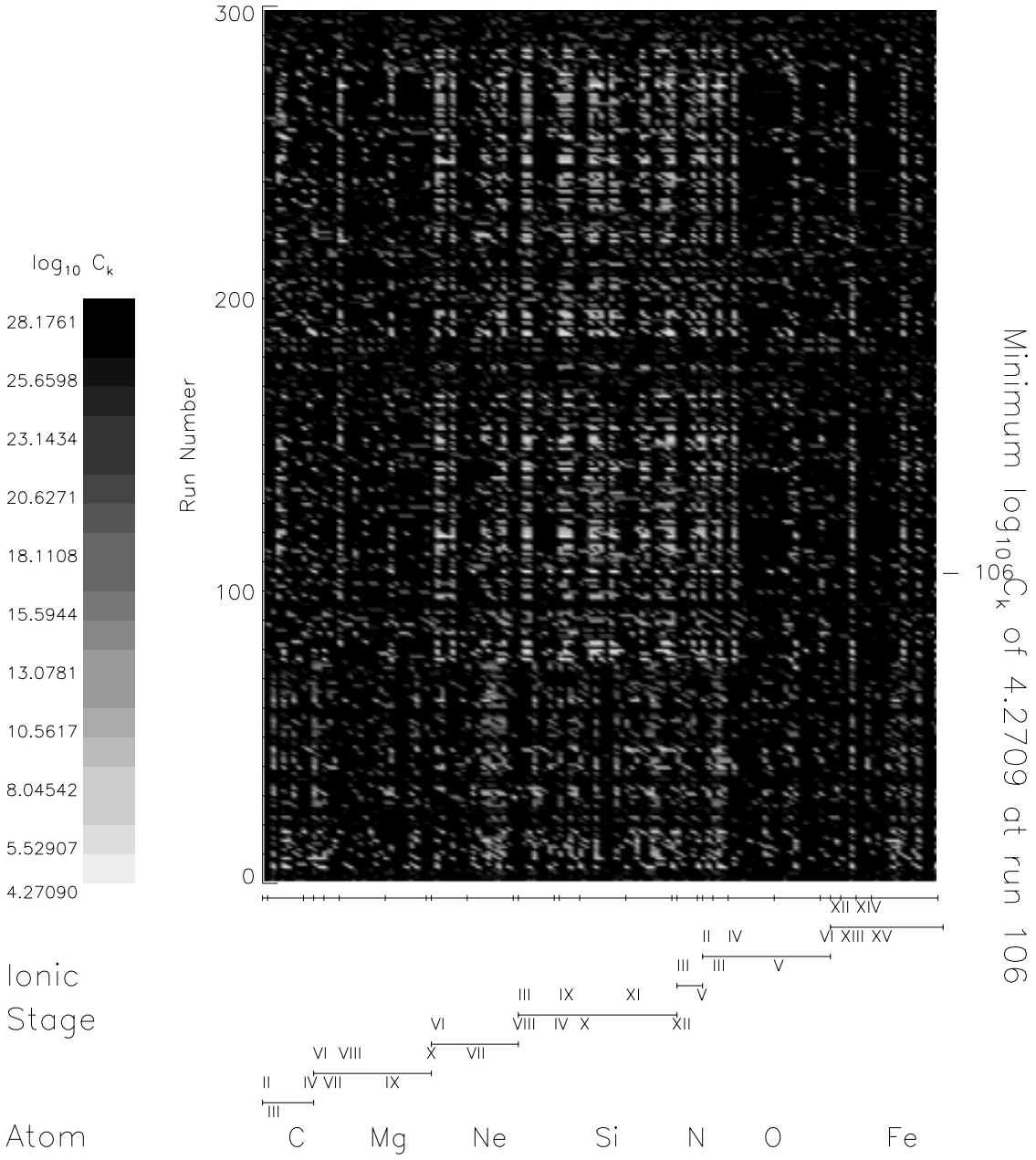

Figure 5.6: This figure identifies the optimal subsets of the 133 emission lines used (grouped according to their atomic/ionisation stage) at the end of each of the 300 Monte Carlo runs of SELECTOR and simultaneously highlights the scale of the combinatorial problem. The runs are sequenced from bottom to top and each run is colour coded with the lowest condition numbers (an attribute of the lines selected) appearing white and increasing in darkness as the condition number of the run increases. Run 106 (indicated on the right) has obtained a condition number of $\log_{10} C'_K = 4.2709$, considerably lower than that ($\log_{10} C'_K = 11.55$) for the "all-lines" approach.

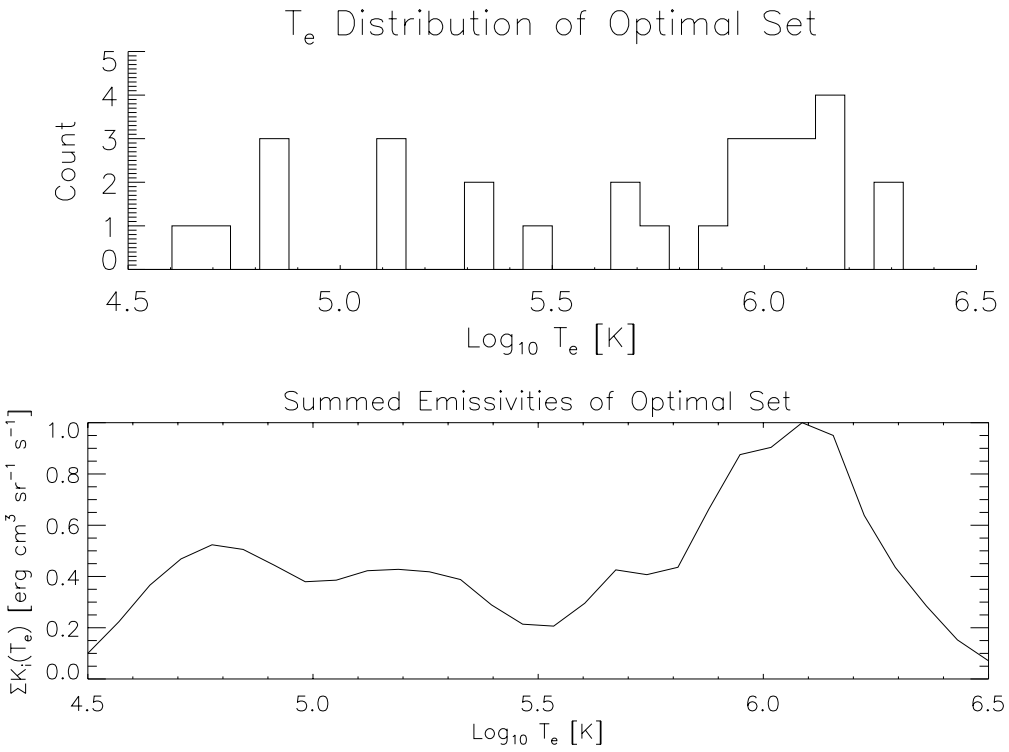

Figure 5.7: The upper plot shows the distribution of emissivity maxima for the lines in Table 5.3. The lower plot shows the normalised summed emissivities $\left(\widehat{S_{5000}(T_e)}\right)$. These emission lines all belong to the optimal subset of run 106.

Table 5.3: Details of the optimal subset of emission lines only. These are the lines belonging to run 106 that form a kernel matrix with $\log_{10} C_K = 4.2709$. Given are the ions to which the line belongs, wavelengths λ (Å), the number of times the particular line was selected and the temperature at with the emissivity of the line peaks T_e^{max} (K).

Ion	λ (Å)	Count	$\log_{10} T_e^{max}$ (K)	Ion	λ (Å)	Count	$\log_{10} T_e^{max}(K)$
C III	1175.59	44	4.8	C III	1176.36	62	4.8
Mg VII	431.188	48	5.7	Mg VIII	763.184	50	5.8
Mg IX	439.176	38	5.9	Mg IX	445.980	45	5.9
Mg IX	706.060	46	5.9	Mg X	609.793	108	6.0
Ne VII	559.948	43	5.6	Ne VII	887.279	119	5.6
Si III	1298.94	48	4.7	Si IX	674.650	43	6.0
Si IX	344.951	74	6.0	Si X	287.092	117	6.0
Si X	356.050	70	6.0	Si X	624.729	66	6.0
Si XI	368.378	161	6.1	Si XI	565.578	67	6.1
Si XI	582.886	195	6.1	Si XI	371.609	37	6.1
N III	991.502	140	4.8	O II	539.085	81	4.6
O IV	1397.23	100	5.1	O IV	1399.78	107	5.1
O IV	1404.80	114	5.1	O V	760.227	54	5.3
O V	1213.80	102	5.3	O VI	150.089	133	5.4
Fe XV	171.839	34	6.2	Fe XV	303.048	44	6.2

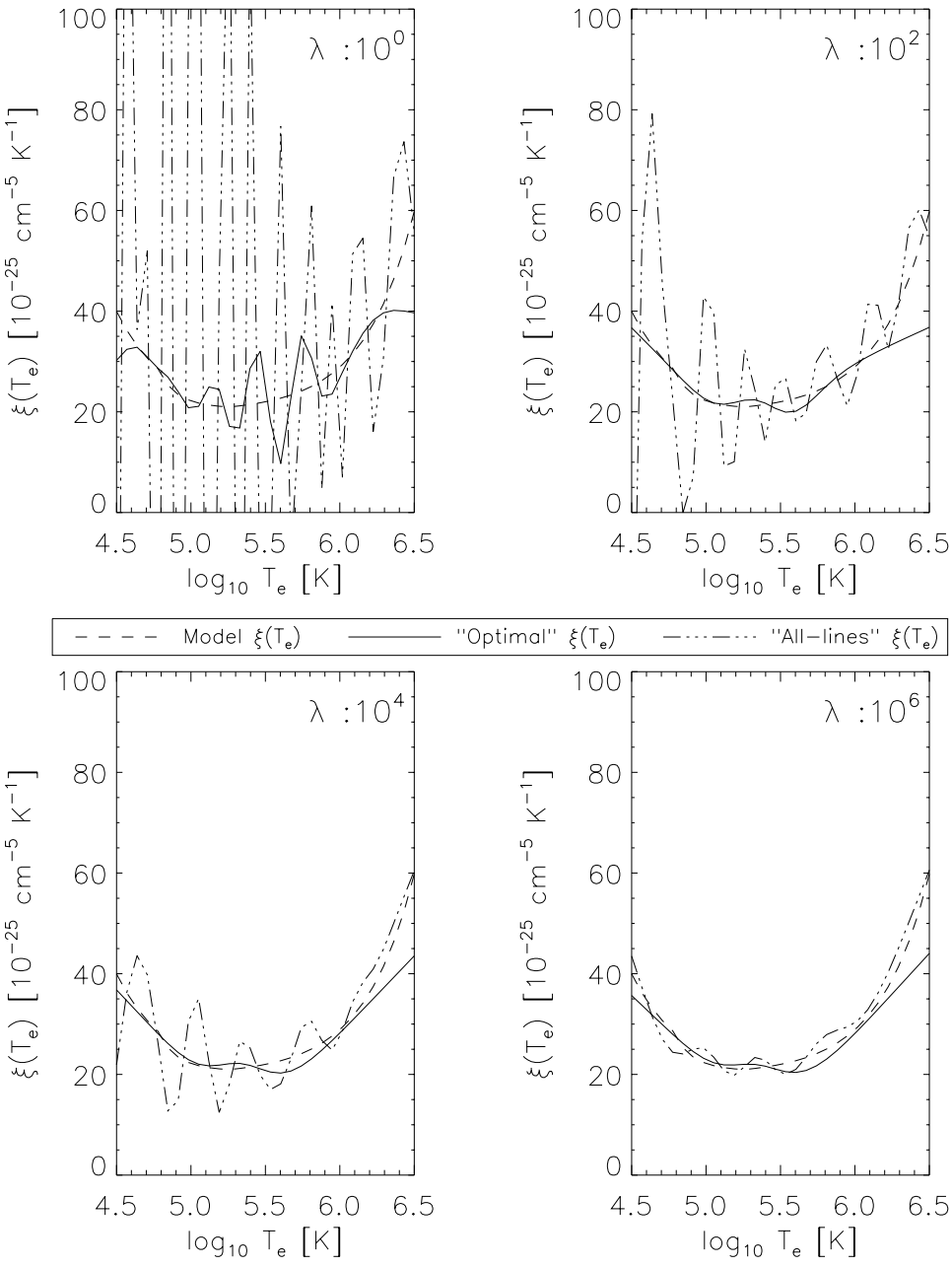

Figure 5.8: The regularised inversion (for smoothing parameters (λ) varying from 10^0 to 10^6) of line intensities calculated for a model $\xi(T_e)$ function (dashed line). The $\xi(T_e)$ function recovered (solid line) is clearly more numerically stable than that for the "all-lines" (dot-dash line) approach in the presence of errors in the line intensities. The line intensities used in these inversions have normally distributed errors of 15%. Error bars on the solutions are not given so not to overcrowd the plots.

or density gradient weighted mean square electron density is a quantity that *would* be of great value if it was possible to infer $\zeta(n_e)$ reliably from the observed spectra.

We have already seen that the inference of such a distribution requires the solution of a Fredholm integral equation of the first kind. Assuming this time that the plasma volume under observation is isothermal, say with $T_e = T_o = 10^5$ K, we can reduce equation (5.1) for the total intensity of line l, for emissivity $K_l(n_e, T_o) = K_l(n_e)$, to

$$I_l = \int_{n_e} K_l(n_e)\ \zeta(n_e)\ dn_e\ . \tag{5.15}$$

This inverse problem is significantly different from that discussed above, primarily because of the functional behaviour of the line emissivities with electron density. One of the important features recognised above was that the emissivity of each line as a function of temperature is well approximated by a Gaussian, however functions $K_l(n_e)$ are *not*. Indeed, the majority of K_ls are 'broad', flat functions covering the entire density domain of interest $n_e(10^8 - 10^{12}$ cm$^{-3})$.

Given that the above statement is true we are faced with a different hurdle from that of the previous section. If the emissivities of the lines are 'flat' in the functional sense then there will be an increase in row linear dependence and a corresponding increase in C_K compared with the $\xi(T_e)$ problem. The $\zeta(n_e)$ inverse problem is extremely poorly conditioned in comparison. So, as was the case in the discussion above we have to ask what properties of the line emissivities will distinguish them from the rest such that the conditioning is improved? Under the assumption that the emitting volume of solar plasma we are modelling is isothermal (at some temperature T_o) we consider the form of $K_{res}(n_e)$ and $K_{int}(n_e)$. At all electron densities $K_{res}(n_e)$ essentially corresponds to the dependence of the radiating element's abundance relative to that of hydrogen and will decrease monotonically with increasing n_e. Again, the form of $K_{int}(n_e)$ depends on the critical density (see Section 2.2) and can be categorised as follows :

- At low densities, radiative decay dominates and the emissivity is essentially constant.

- At densities around the critical density ($n_e \approx \frac{A_{12}}{C_{23}}$) the radiative and collisional mechanisms compete and the result is an emissivity varying as n_e^{δ}, where $(-1 < \delta < 0)$ depending on the atom.

- At high electron densities collision processes dominate and the metastable level will attain a Boltzmann equilibrium and the emissivity will vary as n_e^{-1}

One main exception to these rules exists. For lines excited from low-lying metastable levels their emissivities will vary as n_e^δ $(0 < \delta < 1)$ as the population of the metastable level becomes comparable to that of the ground level, eventually attaining a Boltzmann where the emissivity will become constant. Figure 5.9 shows the functional behaviour of the two lines of N IV discussed previously (765.147, 1486.496 Å) with n_e.

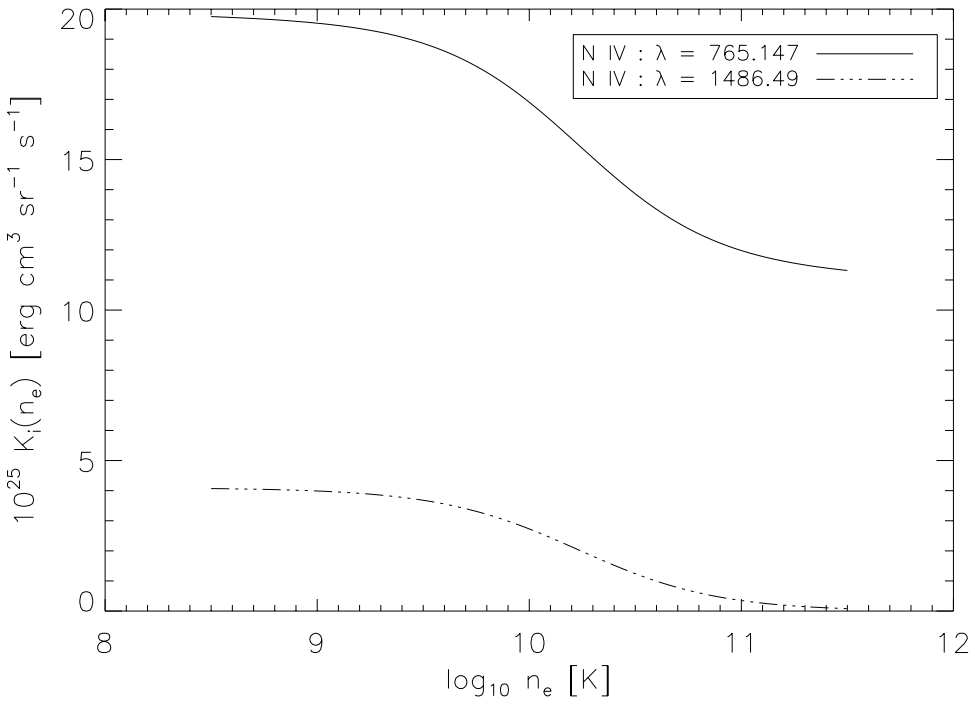

Figure 5.9: The emissivities of a resonance line (solid line) and intersystem line (dashed line) as functions of electron density only. These are for lines of N IV (wavelengths 765.147, 1486.496 Å) calculated for an assumed isothermal plasma of $T_o = 10^5$ K.

Where, in the $\xi(T_e)$ problem, the emissivities were roughly Gaussian functions of temperature and we could easily justify the choice of lines, e.g., by the kernel 'evolving' towards the identity matrix. We now have relatively featureless emissivities. The only real feature they display are gradients of order $\delta n_e^{\delta-1}$ at particular points in the n_e domain. Considering this, even though the emissivities have gradients at specific densities they will essentially be scalar multiples of one another in the 'flat' regions. This confirms that there will be a high degree of linear dependence when (or if) these lines are selected to be in the kernel matrix together.

The degree to which the condition number depends on linear dependence in the kernel matrices presents a new problem. The matrices become increasingly singular and can have a number of *zero* singular values. This is a situation best avoided and requires that we

re-address the way in which SELECTOR calculates C_K. Apart from the condition number estimate of Cline et al. (1979), there is another way to estimate C_K that will give consistent results and we will use this estimate for these results. A clue is in figure 5.2. Observe that the gradient of the singular value distributions for the "all-lines" $\zeta(n_e)$ and $\xi(T_e)$ kernels are different and that the gradient of the $\zeta(n_e)$ case is significantly greater. Similarly, we see that the condition number of the $\zeta(n_e)$ "all-lines" kernel is very much higher than that of the $\xi(T_e)$ case ($\log_{10} C_K = 17.04$ as opposed to 11.55). It is trivial to obtain a algebraic expression for this "relationship". On fitting a straight line (with equation $y = mx + c$) through the logs of the first Q singular values (i.e. the non-zero ones) we see that

$$C'_K \approx 10^{-mQ} \tag{5.16}$$

with $C'_K = C_K$ exactly when $Q = M$ (M is the number of singular values). On making this simple addition to the code of SELECTOR and fixing $Q = 25$ we will again investigate the results of 300, 5000 generation, runs to identify the set of emission lines that minimises the condition number of the $\zeta(n_e)$ inverse problem. Given that, on performing a SVD on the $\zeta(n_e)$ "all-lines" kernel yields a value of $\log_{10} C_K = 17.05$, the gradient method described above gives $\log_{10} C'_K = 16.02$ (for $Q = 25$).

We are looking for a subset of these lines with considerably lower value of C'_K. Figure 5.10 shows the results of the ensemble of 300 runs of SELECTOR and identifies, in a more striking way, a subset of lines selected more than the mean of 104.436 selection frequency. Similarly, figure 5.11 shows the variation of the selection with each run color-coded to correspond to the condition number estimate C'_K defined by equation (5.16). It is clear that run 285, $\log_{10} C'_K = 9.2116$, when compared to that of the "all-lines", contains a subset of the lines which has a lower value of C'_K. However, it is also clear from the mottled pattern of figure 5.11 the difficulties of selecting such an optimal set when virtually all sets are very poorly conditioned. This mottling may be an artifact of the estimate used to calculate C'_K. Tables 5.4 and 5.5 show the combination of the results presented in figures 5.10 and 5.11. Table 5.4 gives the details of all the lines selected 25% greater than the mean selection frequency. The lines belonging to the subset of produced by run 285 ($*$), values of $\left|\frac{dK(n_e)}{dn_e}\right|$ and the n_e^* (the electron density at which $\left|\frac{dK(n_e)}{dn_e}\right|$ is greatest) to help obtain a physical description of why this particular set of lines is chosen above the others. Similarly, Table 5.5 presents the details of only the lines selected in run 285 and figure 5.12 shows the coverage of the selected line emissivities (normalised), $\widehat{S_{5000}(n_e)}$. One feature very evident in this figure is the flatness of the summed

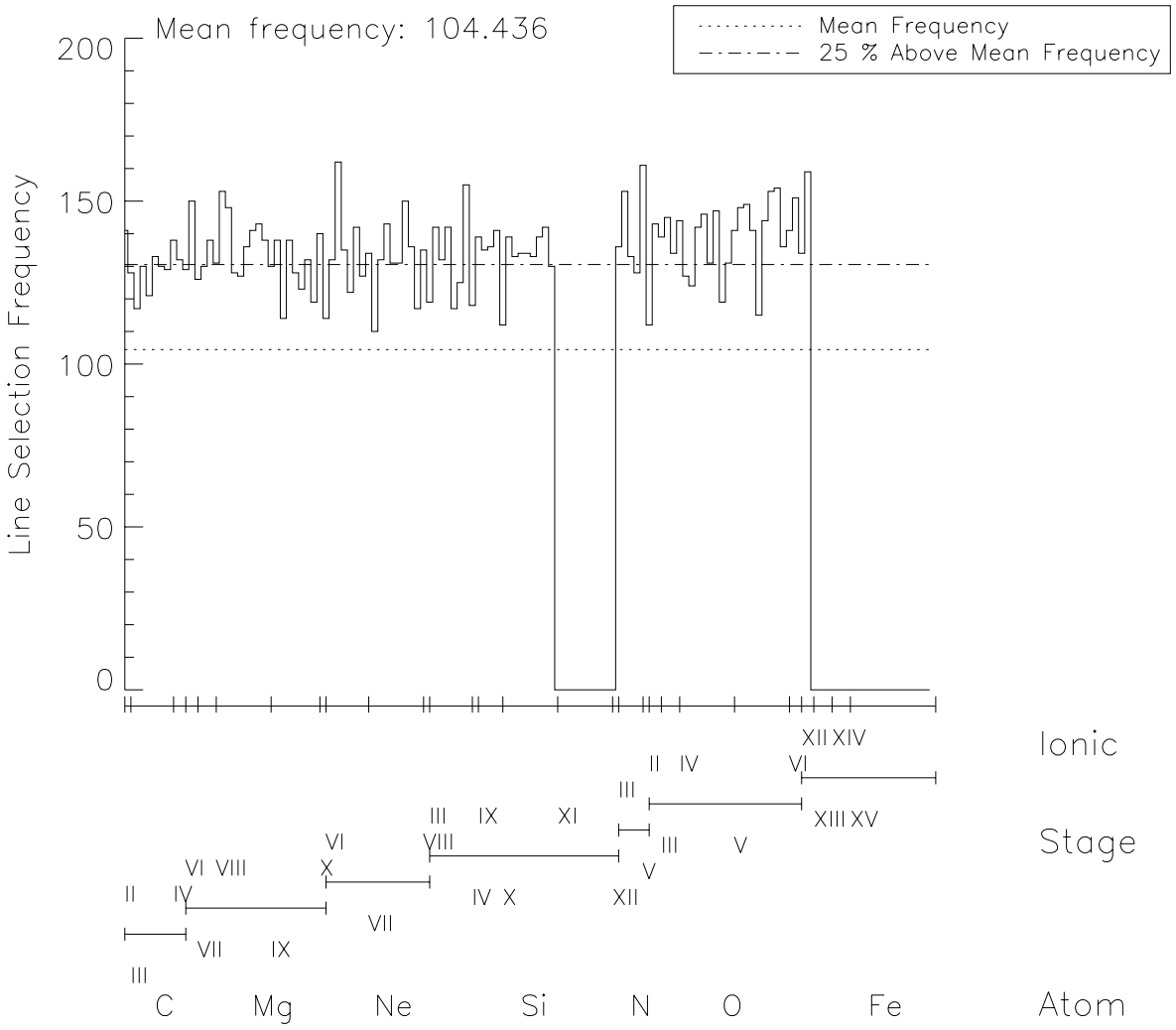

Figure 5.10: Monte Carlo histogram of line selection frequency versus line results for 300 runs of the SELECTOR. This clearly shows the existence of a subset of the 133 lines that have selection frequencies significantly greater than the mean of 104.436. The axes are labelled as in figure 5.5

emissivities.

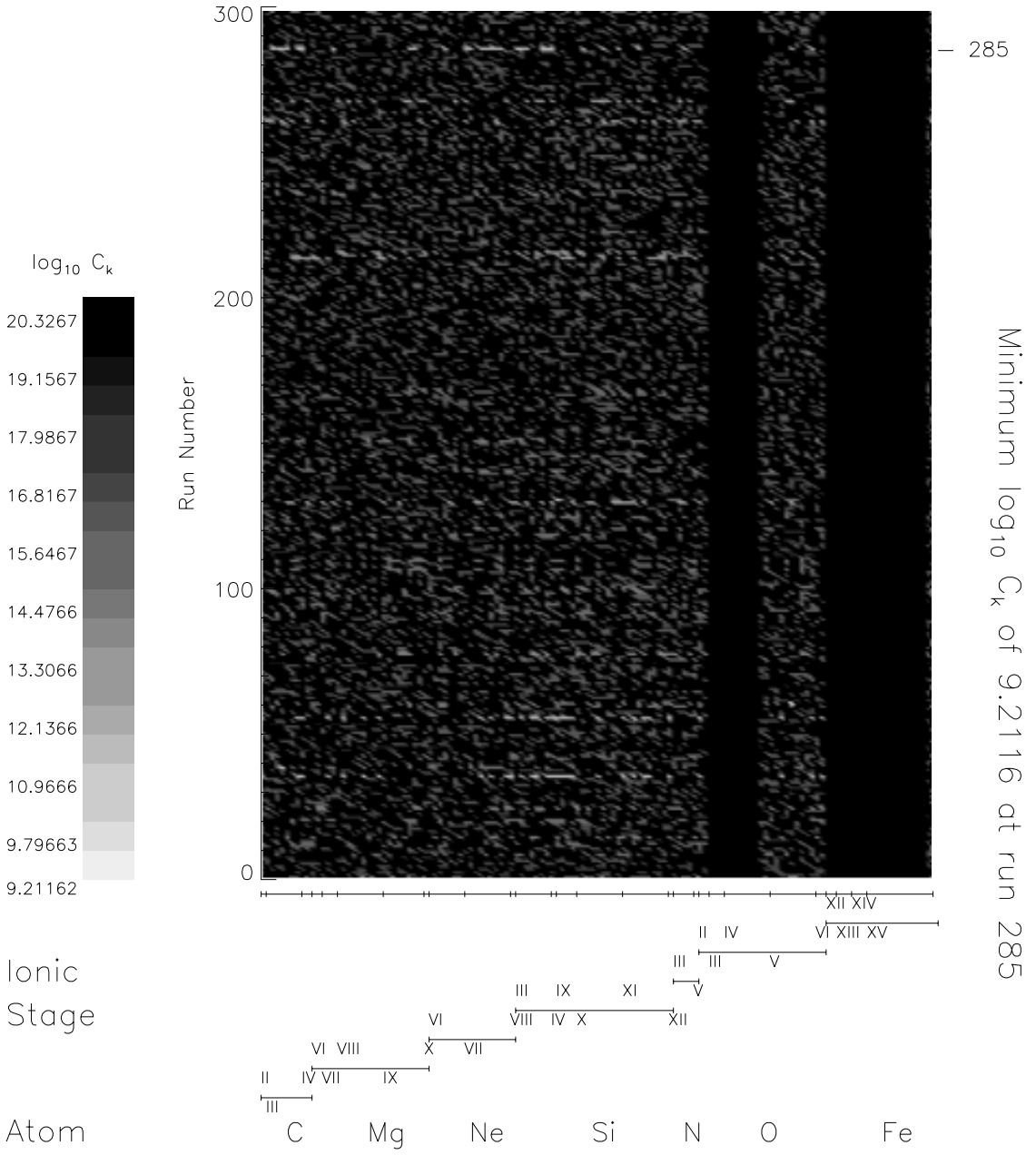

Figure 5.11: This figure identifies the optimal subsets of the 133 emission lines used (grouped according to their atomic/ionisation stage) at the end of each of the 300 Monte Carlo runs of SELECTOR and simultaneously highlights the scale of the combinatorial problem. The runs are sequenced from bottom to top and each run is colour coded with the lowest condition numbers (an attribute of the lines selected) appearing white and increasing in darkness as the condition number of the run increases. Run 285 (marked on the right) has obtained a condition number of $\log_{10} C'_K = 9.2116$, considerably lower than that ($\log_{10} C'_K = 16.02,\ Q = 25$) for the "all-lines" approach.

Table 5.4: Details of the emission lines selected most at the end of the 300 Monte Carlo 5000 generation runs of SELECTOR. The emission lines included here are those with selection frequencies 25% greater than the mean of 104.436 counts. The lines indicated by an asterisk (∗) are those belonging to run 285, the set having the minimum value of $\log_{10} C'_K = 9.2116$. Also given are the ions to which the line belongs, wavelengths λ (Å), $\left|\frac{dK(n_e)}{dn_e}\right| (= |K'|)$ and the value of n_e^*, the value where the emissivity gradient is greatest. Values n_e^* of 8.8 or 11.2 indicate that maximum occurs between that value and the appropriate limit of the density domain ($8 \leq \log_{10} n_e \leq 12$).

Ion	λ (Å)	Count	$\lvert K'\rvert$	$\log_{10} n_e^*$		Ion	λ (Å)	Count	$\lvert K'\rvert$	$\log_{10} n_e^*$	
C II	1335.66	141	0.1	8.8		C IV	312.420	138	4.0	8.8	∗
Mg VI	1190.07	150	0.0			Mg VII	431.188	138	5.1	10.2	∗
Mg VIII	436.671	153	1.9	11.2		Mg VIII	782.913	148	2.3	10.5	
Mg VIII	789.964	136	8.5	11.2		Mg VIII	772.749	141	4.0	11.2	
Mg VIII	355.998	143	8.3	11.2		Mg VIII	352.460	138	2.9	11.2	
Mg IX	448.293	138	8.0	11.2		Mg IX	439.176	138	9.3	11.2	
Mg X	609.793	140	6.2	9.4		Ne VI	454.072	162	1.7	11.2	
Ne VI	562.711	135	0.0		∗	Ne VI	1006.09	142	0.0		
Ne VII	895.175	143	6.8	11.2	∗	Ne VII	465.220	150	2.6	11.2	∗
Ne VII	887.279	136	9.1	8.8	∗	Ne VIII	770.408	135	3.5	11.2	
Si III	1206.49	142	1.9	8.8	∗	Si III	1296.72	142	0.0		
Si III	1298.94	155	0.1	8.8	∗	Si IX	950.082	139	1.1	11.2	
Si IX	692.731	135	6.5	11.2		Si IX	674.650	136	2.8	11.2	
Si IX	344.951	141	4.2	11.1	∗	Si X	611.658	139	1.5	11.2	
Si X	624.729	139	2.2	11.2		Si X	649.268	142	3.4	11.2	
N III	771.544	136	0.7	9.6	∗	N III	991.502	153	6.5	10.1	
N V	1238.82	161	26.6	10.6		O II	539.085	143	0.0		
O III	833.715	139	78.5	10.9		O III	1666.14	145	49.4	10.5	
O IV	1397.23	144	3.7	10.1		O IV	625.127	142	13.6	11.1	
O IV	624.618	146	6.8	11.1		O IV	1399.78	147	15.8	11.2	
O V	761.128	141	0.9	11.2		O V	760.227	148	0.8	11.2	
O V	760.446	149	3.9	11.2		O V	758.676	141	1.3	11.2	
O V	759.441	144	1.0	11.2		O V	629.732	153	25.1	11.2	
O V	1213.80	154	0.0			O V	1218.34	136	3.0	10.3	∗
O VI	150.089	141	5.0	11.2		O VI	1031.91	151	0.0		

Table 5.5: Details of the emission lines belonging to run 285, the set having the minimum value of $\log_{10} C'_K = 9.2116$. Also given are the ions to which the line belongs, wavelengths λ (Å), $\left|\frac{dK(n_e)}{dn_e}\right| (= |K'|)$ and the value of n_e^*, the value where the emissivity gradient is greatest. Again, values n_e^* of 8.8 or 11.2 indicate that maximum occurs between that value and the appropriate limit of the density domain ($8 \leq \log_{10} n_e \leq 12$).

Ion	λ (Å)	Count	$\lvert K'\rvert$	$\log_{10} n_e^*$	Ion	λ (Å)	Count	$\lvert K'\rvert$	$\log_{10} n_e^*$
C III	977.020	117	4.4	9.1	C III	1175.98	130	10.7	9.1
C III	1175.26	121	11.3	9.2	C III	1176.36	133	14.1	9.2
C III	1174.93	129	13.9	9.2	C IV	312.420	138	4.0	8.8
Mg VII	431.188	138	5.1	10.2	Mg IX	441.199	123	6.9	11.2
Mg IX	368.070	132	7.5	11.2	Ne VI	562.711	135	0.0	
Ne VII	564.528	134	2.9	11.2	Ne VII	561.378	110	1.8	11.2
Ne VII	895.175	143	6.8	11.2	Ne VII	559.948	131	2.4	11.2
Ne VII	562.993	131	8.7	11.2	Ne VII	465.220	150	2.6	11.2
Ne VII	887.279	136	9.1	8.8	Si III	1298.89	119	0.1	8.8
Si III	1206.49	142	1.9	8.8	Si III	1301.14	132	0.0	
Si III	1294.54	125	0.1	8.8	Si III	1298.94	155	0.1	8.8
Si IV	1393.75	118	19.3	8.8	Si IX	344.951	141	4.2	11.1
Si X	356.050	133	3.9	11.2	Si X	292.167	130	5.5	11.2
Si X	611.658	139	1.5		N III	771.900	136	0.7	9.6
O V	762.004	115	1.3	11.2	O V	1218.34	136	3.0	10.3

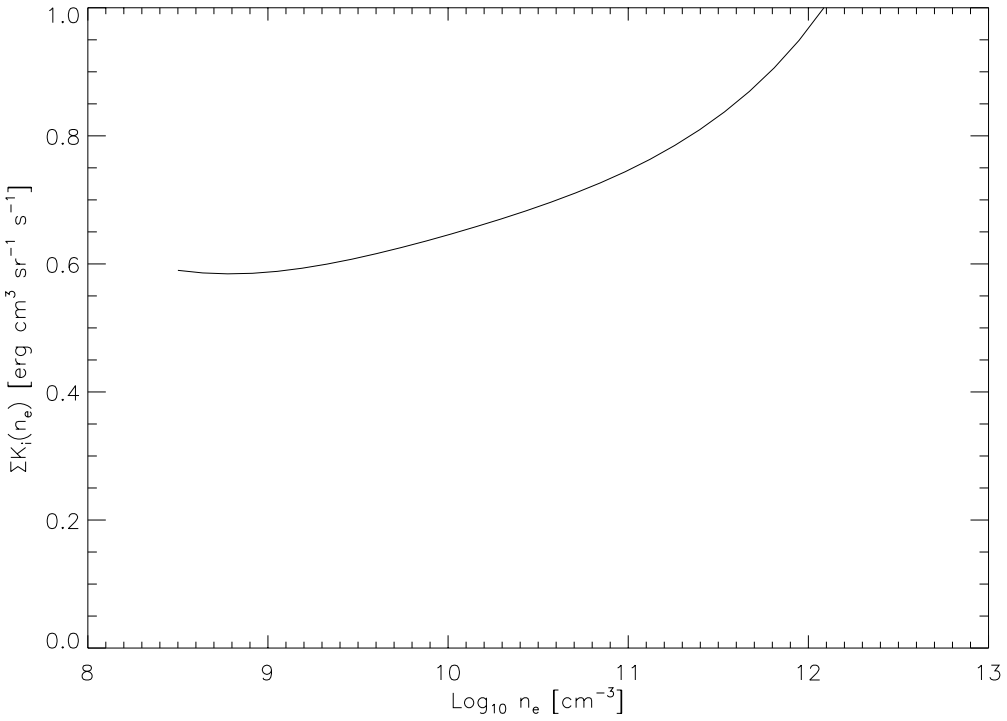

Figure 5.12: A plot showing the summed (normalised) emissivities $\left(\widehat{S_{5000}(n_e)}\right)$. These emission lines all belong to the optimal subset of run 285 given in Table 5.5.

Again, we will seek a regularised solution to the inverse problem to see how stable the solution of the "all-lines" compares to that using the reduced optimal set of run 285. We perform the inversion for a range of smoothing parameters λ $(10^1 - 10^5)$ to obtain a solution to equation (5.15). The calculated line intensities for this model $\zeta(n_e)$ function have normally distributed random errors of 5% magnitude. Figure 5.13 clearly shows that the inversion performed with the optimal subset of lines is significantly more numerically stable than that obtained when using the "all-lines" approach, especially over the range of smoothing parameters used.

The optimal value of C'_K is significantly higher than that of the previous section, as would be expected from inspection of figure 5.2, but because of the estimate used may be slightly inaccurate. The high value of C'_K alone would indicate why $\zeta(n_e)$ is not a 'popular' diagnostic of the emitting plasma (although discussed at length in Almleaky et al. 1989 and Brown et al. 1991); the numerical instability and non-uniqueness of the inferred solution and would make derivation of the physical mechanisms for the radiating plasma useless. On inspection of figure 5.13 such an argument is further reinforced, considering especially that the calculations presented were made using line intensities with only 5% errors. This would

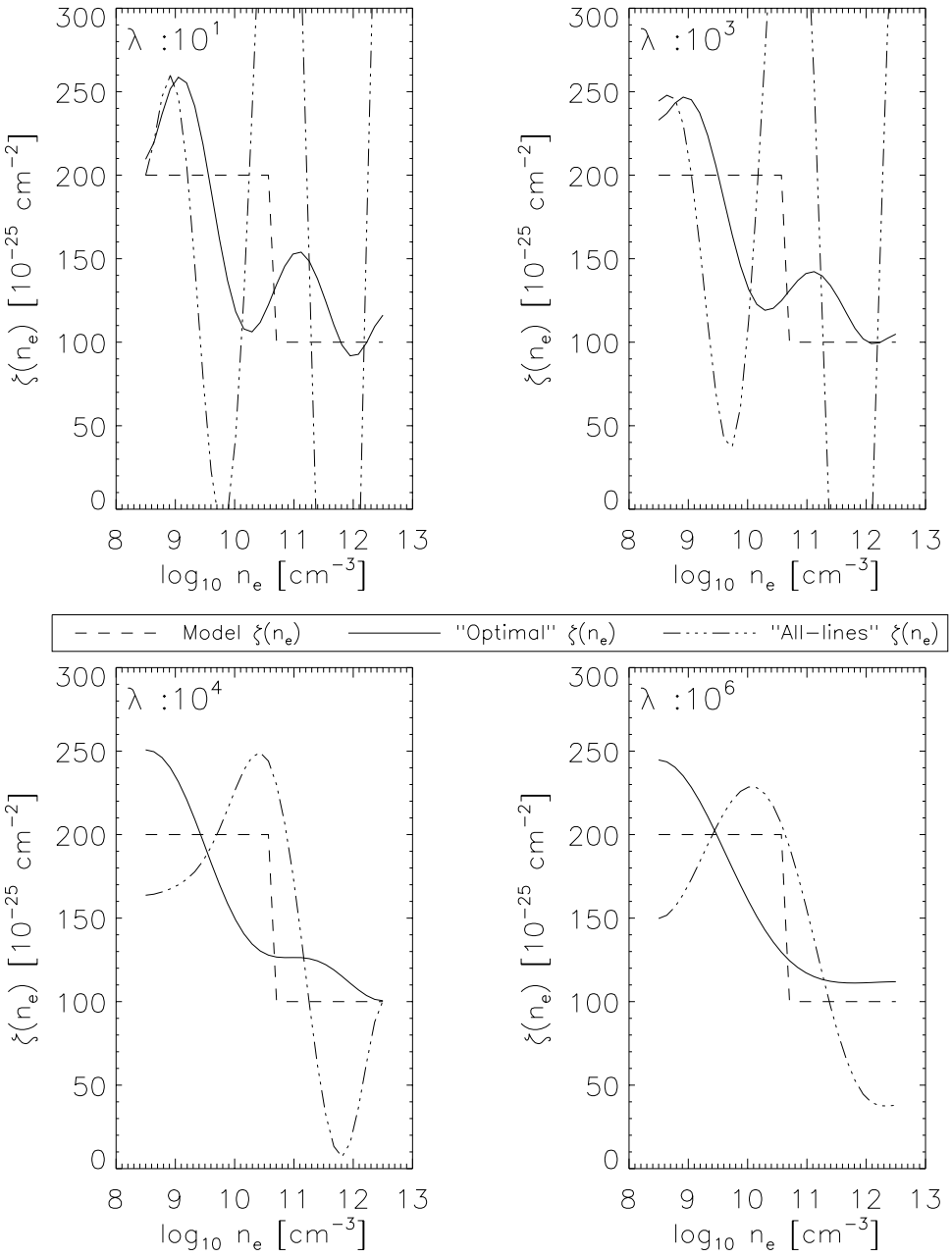

Figure 5.13: The regularised inversion (for smoothing parameters (λ) varying from 10^1 to 10^5) of line intensities calculated for a model $\zeta(n_e)$ function (dashed line). The $\zeta(n_e)$ function recovered (solid line) is clearly more numerically stable than that for the "all-lines" approach in the presence of errors in the line intensities. The line intensities used in these inversions have normally distributed errors of only 5%.

be an optimistic lower bound on the error estimate. This means that the outlook is bleak as these inferred solutions are possibly the best we can recover from this data. These numerical problems still occur even when using the optimal set of lines. Hence the reason why many density diagnostics are acquired using the line-ratio method mentioned in Chapters 2 and 4 and not using $\zeta(n_e)$.

5.4 Discussion

For the trials shown we have established that, for the set of emission lines in the SOHO CDS/SUMER wavelength range, there are subsets which minimised the conditioning problems associated with the inference of plasma characteristic distributions $\xi(T_e)$ and $\zeta(n_e)$. Also, these subsets of emission lines which, when used to infer the emitting plasma structure will yield results of a less ambiguous nature.

Chapter 6

Summary and future work

This Chapter

In this chapter we draw all our arguments and threads together to show that the recovery of the solar physical structure from UV/EUV emission line spectra is no easy task and must be treated with due care and attention.

The importance of obtaining good (useable) distributions for solar plasma diagnostic quantities is paramount if we are to unlock the mysteries surrounding the coronal, chromospheric and flare heating problems (reviewed recently; Zirker 1993). The ability to support observationally certain mechanisms relative to others requires that we have, at least, an unique model for the emitting plasma. In this thesis we have developed, using new and what some may class as unconventional, methods with an open-minded perspective to do just that. We have used an approach that determines the underlying plasma characteristics to a higher degree of numerical stability and uniqueness than previously obtained. The argument of this thesis from the outset can simply be expressed as (McIntosh 1998 - Oral Presentation)

> "If we are to learn anything about the solar atmosphere from the SOHO and similar missions we *have* to use data extraction methods which are most robust and accommodate *all* the errors likely to occur. Such methods will, in return, increase the reliability and uniformity of results inferred using these methods."

We have systematically introduced construction methods for inferring unique distributions from observed UV/EUV optically thin line emission spectra. In the main, particular emphasis is placed on the wavelength range observable with the Coronal Diagnostic Spectrometer (CDS;

150 - 800 Å) and Solar Ultraviolet Measurement of Emitted Radiation (SUMER; 780 - 1610 Å) instruments in the SOHO payload.

The vast majority of the technical details required to understand the work contained in this thesis are laid down in Chapter 2. In Section 2.1 we gave a discussion of the general details to aid in the understanding, perception and solution of inverse problems. We noted that it was important to formulate the inverse problem correctly and, for the solar inverse problem considered predominantly in this thesis, we require a basic knowledge of the atomic mechanisms of highly ionised species present in the solar atmosphere. Theoretical knowledge of the atomic "zoo" we call the Sun allows us to diagnose the current state of the emitting plasma structure and, in Section 2.2, we discussed these diagnostics, their formulation and possible errors resulting in their use.

Chapter 3 introduced, at an elementary level, the basic framework from which much of the argument of this thesis is constructed. There we introduced the terminology, mechanism and adaptability of Genetic Algorithms (GA). As a test of the flexibility of a GA we apply it as a method for obtaining 'unbiased' decompositions of emission line spectra. We have demonstrated that a GA is a robust, and effective, optimisation method for navigating potentially hazardous solution/parameter spaces and that Ga-GA (the Gaussian fitting GA) demonstrated the ease with which *a priori* constraints can be applied to the data under analysis. In addition to the ease with which constraints can be placed on the data under analysis we have provided evidence that Ga-GA provides a more accurate spectral decomposition (especially at the limit of instrumental resolution) than standard decomposition algorithms. It is also clear from the analysis presented therein that there is no such thing as a free lunch; any advance in accuracy must be accounted for by significant increase in the time taken to obtain that accuracy and would limit Ga-GA's effectiveness as an on-line analysis tool, e.g. for analysing simple line profiles (single or double) in tokamac plasmas. However, as is true in many astronomical cases, the time taken to run the algorithm is irrelevant compared to the accuracy required of the decomposition. As a thought example consider the decomposition of a SUMER quiet Sun spectrum, like that presented in figure 3.8. A mis-calculation of the line width or shift relative to the 'known' laboratory wavelength would be enough to completely discount the "nanoflare" model of Parker (1988) and the magnitude of observed downflows in the transition region emission lines (Wikstol et al. 1997; Wikstol et al. 1998) of O IV and S IV ($1398 \leq \lambda \leq 1402$ Å) that can be inferred from these measurements.

Section 4.1 endeavours to break down the conceptual walls that have developed since the dawn of space borne UV/EUV spectroscopy in the early 1960s. Here we make clear reference to the two schools of thought that presently exist that are vying for 'control' of this particular avenue of solar physics; each professing the correctness of their approach.

The earliest (effectively zeroth order) 'line ratio' approach actually harks back to the work on planetary nebulae by Menzel et al. (1941). It uses a zeroth order approach to obtain 'mean' values of the observational spectroscopic quantities n_e and T_e. These estimation methods have been shown to be highly ambiguous (Almleaky et al. 1989) because they assume that the emitting plasma volume (solid angle) subtended by the spectrometer slit is isothermal, or homogeneous in n_e, or both. This is clearly not a valid assumption as can be seen by looking at any image of the Sun's atmosphere. It does however allow order of magnitude estimates of n_e and T_e to be made within the degree of uncertainty in the measurement and within the quality of the theoretical atomic coefficients used. Although, as noted in Chapter 4, the addition to the analysis of more line ratios can establish limiting values for n_e and T_e of the emitting volume (Brown et al. 1991).

The first order approach, to obtain distributions of these temperatures and densities in the emitting volume, was first expressed by Pottasch (1964) but generalised by Craig & Brown (1976) to account for the multiple regions of differing temperature (and density) along the line of sight in the emitting volume. This second, differential emission measure (DEM), approach has been well documented as a more rigorous, in a mathematical sense, method of obtaining such characteristic distributions. The identification of the DEM approach as requiring the solution of a Fredholm integral equation for the relation governing the formation of noisy emission line intensity data (Craig & Brown 1976) was made following the work of Jefferies et al. (1972a, b). Recently however, Judge et al. (1997) stated that this 'inverse' DEM method is also fraught with instability caused by uncertainties in the atomic parameters required for the calculation and not just the numerical difficulties involved in performing the inversion itself (Craig & Brown 1986) for random data noise. We have addressed both of these approaches, aspects of error sources and also we have been able to produce direct relationships between a set of mean spectroscopic quantities and the corresponding DEMs of temperature ($\xi(T_e)$; Section 4.1.1), density ($\zeta(n_e)$; Section 4.1.2) and the generalised form of the bivariate DEM ($\mu(n_e, T_e)$; Section 4.1.3) for situations in which all these quantities are meaningfully defined. These relations show that the road linking the two schools of thought is not as arduous as may be inferred from the current literature. Indeed, we have shown that

the methods are precisely equivalent.

It was soon clear that the main benefits of the line ratio like approach coupled with the full mathematical rigour of solving the Fredholm integral equation would give a means of avoiding the systematic errors in the atomic calculations detailed in Section 2.2.3 and obtaining meaningful characteristic distributions. This alone is a great advance since the effect of the line ratio method on these irregularities circumvents the instability imposed on the inferred solution of the direct integral inversion (Judge et al. 1997). Section 4.2 sees the introduction of the Ratio Inversion Technique (RIT) and in Section 4.2.2 we have discussed, in detail, how this GA based routine solves the highly non-linear optimisation problem

$$\chi^2 = X^2(R_{obs}, R_{calc}) + \lambda\Phi(f(s_e))$$

with respect to a "smooth" solution $f(s_e)$ (occurring in the calculation for R_{calc}, the line ratios generated to match the observed ratios R_{obs}). We have shown in the sections following Section 4.2.2 that the RIT provides a solution of unprecedented stability in the recovery of the univariate plasma DEM functions $\xi(T_e)$ and $\zeta(n_e)$ compared to a standard inversion algorithm (cf. the GUIPS routines of Section 2.1). The value of this result is most strongly emphasised when considering the realistic estimates of uncertainties in the line emissivities as large as ($\sim 100\%$). It is clear then that if the solution is stable to errors in the atomic calculations used, as well as being numerically stable, that we can place greater store in the resulting analysis of those DEM functions to form atmospheric models and the like from the UV/EUV line spectra.

It is all well and good to test the RIT in the ideal conditions described in Chapter 4. However, Section 4.4 places a new obstacle to test the 'initiative' of the RIT. Here we present results of the RIT operating on spectra obtained by the SERTS-89 rocket. The wavelength coverage of the SERTS spectra is $170 - 450$ Å and this particular active region spectrum has been studied in detail by Thomas & Neupert (1994) and more recently by Lanzafame et al. (1998). We demonstrated that the RIT uncovers features in the various $\xi(T_e)$ functions not observed in these previous studies and that our proposed methods of scaling, smoothing and choice of the optimal smoothing parameter are accurate. These results highlight the basic inherent problem mentioned above, the need for a uniform approach (same data - same DEM) to these inverse problems like that discussed in Harrison & Thompson (1991). Also highlighted is the basic ill-posedness of the DEM inverse problems: many forms of solution fit the data and we must acknowledge now that we *cannot* discount any. Indeed, it may well be

that *all* we can achieve with any degree of certainty is to put a vague boundary in the region of the DEM solution spaces where we would expect possible DEM functions for various solar regions (active and quiet) to lie.

Chapter 5 sees a slight change of tack. We investigated a question first posed in Craig & Brown (1976)

> "What makes particular UV emission lines better than others in terms of recovering the plasma source function ?"

Again, we have used a GA based tool to investigate the factors controlling the numerical stability of inverse problem solutions in the presence of data noise, i.e. limiting the effect of poor conditioning. Essentially we have isolated subsets of emission lines that substantially reduce the response of the integral inversion to considerable data noise. In Section 5.2, for $\xi(T_e)$, we reduced the degree of poor conditioning by minimising the condition number from 10^{11} to $\sim 10^4$ by careful choice of the emission lines we use in the inversion process itself. Likewise, in Section 5.3 we have isolated the corresponding set of emission lines for the univariate $\zeta(n_e)$ inverse problem which reduce the condition number from 10^{17} to $\sim 10^9$. From these results we see that careful consideration of the lines analysed can yield much more numerically stable solutions to a standard DEM inversion.

To summarise, we have clearly shown that the methods employed in this thesis establish a greater degree of uniqueness and numerical stability in the inferred DEM functions which is a positive contribution, particularly in terms of the further interpretation of solar spectroscopic data in uncovering the true mechanism(s) responsible for regenerating and heating the upper solar atmosphere.

6.1 Future Work

The importance in terms of atmosphere modelling make the discussion of this thesis very timely. There are many avenues left to explore with the methods discussed within but particular effort should be made to extend the SELECTOR and RIT methods to allow the further study of $\mu(n_e, T_e)$ which is the "holy grail" of UV inverse spectroscopy, though, because of its physical abstraction, it is the principal subject of only four pieces of literature to date, as far as the author is aware.

Extension of the work presented in Chapters 4 and 5 to study the bivariate DEM $\mu(n_e, T_e)$ will revolve around the re-indexing (transformation) performed in Judge et al. (1997) and

used in Section 4.1.3. The operation of the RIT in recovering the functional form of $\mu(n_e, T_e)$ will be limited by two factors, they are:

1. The inversion mesh will be limited in dimension because when the parameter string used to describe the solution ($30 \times 30 = 900$ elements for a 30 by 30 mesh) is far too long for analysis as a gene string or genotype. This will certainly require greatly enhanced genetic operators (see the discussion of Section 3.4) to proceed.

2. The use of this mesh itself poses another problem which may indeed reduce the pressure on point 1. The number of parameters in the calculation can be limited by choosing a bivariate functional form for $\mu(n_e, T_e)$ which will allow us to form a series expansion with the parameters in this case being the expansion coefficients. However, the identification of such a functional basis is not easy.

The discretisation of the $\mu(n_e, T_e)$ inverse problem is difficult for SELECTOR to handle since we are still only choosing the optimal set of emission lines from the condition number of the resulting kernel matrix (i.e. does it translate linearly from the $m \times n$ case to the $m \times n \times k$ case even though used in Judge et al. 1997 ?). The main difficulty in this case is actually assessing the validity of the condition number estimate used. However, expansion of SELECTOR for the bivariate inverse problem may help provide the necessary physical link between lines that are "good" in the temperature ($\xi(T_e)$) case but do not feature good density ($\zeta(n_e)$) sensitivity or vice versa.

Also great attention must be paid to the smoothing functional used for the RIT solution of the univariate DEM inverse problems. Implementation of the 'data adaptive smoothing' approach of Thompson (1990, 1991) is another possible advance for the RIT. A working knowledge of the smoothing functional is critical to understand the amount of erroneous variation present in any possible solution. In other words, we want to be able to determine regions in the T_e and n_e domains where the smoothing functional is 'in control' and to make sure that any oscillatory variation in the recovered solution is not an artefact of the inversion process itself. This is a truly difficult task as the reader, by now, will appreciate. We, however, believe that it is possible to couple the analysis of the RIT with error bars that not only reflect possible errors in the line emissivities but the emissivity 'coverage' (at all points in the domain). A similar analysis has been performed in Chapter 5 and may lead the way to obtaining a clear result on this front. This should help us to decide whether or not possible variations in the recovered DEM functions are below the numerical and coverage resolution

limits and hence spurious. An estimate of this effect **may** be given, simply, by considering an error bar $\pm\delta f(s_{e_j})$ in the DEM function whose amplitude depends on the contribution from emissivity $K_i(s_{e_j})$ to each data error δg_i for a point s_{e_j} in the s_e (n_e of T_e) domain like

$$\delta f(s_{e_j}) \leq \frac{\sum_{i=1}^{M} \delta g_i}{N \sum_{i=1}^{M} K_i(s_{e_j})} \tag{6.1}$$

where M and N are the number of observations and domain discretisation points respectively. This *ad-hoc* relationship corresponds to a lower weighting of $\delta f(s_{e_j})$ where the kernel coverage is large and conversely a higher weighting when the region of the s_e domain is poorly sampled in the kernels.

To complete this brief analysis of possible applications of the methods contained in this thesis we must address some important issues. These primarily concern the formulation and application of DEM-like techniques when used to analyse UV/EUV spectra from dynamic or clearly non-equilibrium plasmas of both active and quiet regions of the Sun. In Chapter 2 we stated explicitly the conditions under which we may formulate the equation of total line intensity as a DEM integral equation (see, e.g. equation 2.68). We have to take it for granted that the plasma is optically thin, in ionisation equilibrium, have constant atomic abundances and be in a steady state with all the lines observed being formed in the same emitting volume. Clearly, in regions of the solar atmosphere some, if not all, of these assumptions will break down, as any image will show (see, e.g., Golub & Pasachoff 1997), and such cases are discussed in Judge et al. (1995) and Chae et al. (1997). The question that must be asked then is

> "Can we formulate a DEM-type integral equation for the emitting plasma regardless of its non-equilibrium structure ?"

Well, given that we have established the RIT as an useful diagnostic technique for obtaining DEM functions, possibly the best we can do will involve taking continuous (long temporal T_o duration; cf. figure 6.1) observations of a region of the Sun and analyse for some integrated time scale to even out the fluctuations in intensity. This is done in practice but there has been no analysis of a time variation in the DEM functions over a much shorter time scale (i.e. using time series like those in figure 6.1 and performing the inversion for segments with $t \ll T_o$) than those often presented (this is discussed in Chapter 5.6 of Mariska 1992 for the solar transition region). The object of such a study is the correlation between possible DEM fluctuations and brightenings/dimmings of the observed line intensities. Such a correlation

will also help locate particular regions of n_e and T_e (through $\xi(T_e)$ and $\zeta(n_e)$) where heating events are occurring and when.

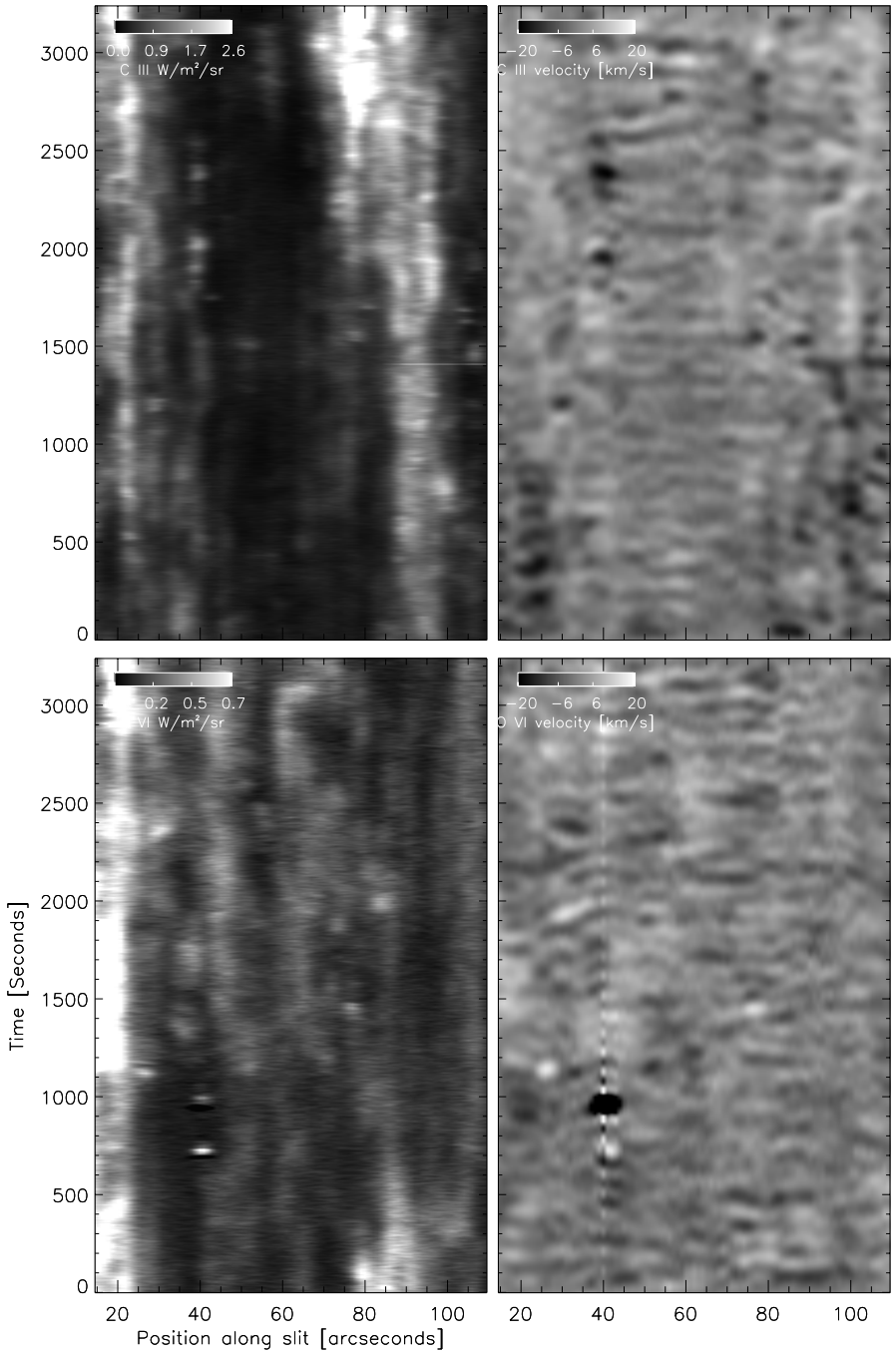

Figure 6.1: SUMER time-series spectra for lines of C III (1175 Å; upper plots) and O VI (1037 Å; lower plots) obtained on October 11$^{\text{th}}$ 1996. The intensity (left) and velocity (right) as functions of position along the spectrometer slit and time. Because these lines are formed at slightly different temperatures, 9×10^4 K and 4×10^5 K respectively, the different behaviour in both spatial and temporal domains of the solar atmosphere is clear over the three thousand seconds of observing time. This figure appears courtesy of Dr. V. Hansteen.

Appendix A

PIKAIA driven Genetic Algorithms

In the following sections we present the full Fortran-77 codes for the PIKAIA (Charbonneau & Knapp 1996) Genetic Algorithm (GA) drivers employed in this thesis. The code for the Gaussian fitting GA (GA-GA) of Chapter 3 is included in Appendix A.1. Also, the code for the Ratio Inversion Technique (RIT), discussed in Chapter 4, is included in Appendix A.2.

A.1 The Gaussian fitting Genetic Algorithm (Ga-GA) code

```
      program Ga-GA
c     -------------

c=============================================================================
c
c     Driver for Gaussian Spectral Decomposition with a Genetic Algorithm
c
c     Scott W McIntosh (V1.01 6 Apr 1998; scott@astro.gla.ac.uk)
c
c  Use : Ga-GA  <data_file> <iseed> <nc>
c
c            data_file     input data file                          [xxx.dat]
c            iseed         random number generator seed (positive) [7187]
c            nc            effective number of gaussian components [1-10]
c
c  I/O Units
c
c       1   input data                       xxx.dat
c      23   final phenotype                  xxx_phenotype.out
c      24   final parameters                 xxx_parameter.out
c      30   fitness of each generation       xxx_fit_gen.out
c      31   parameters at each generation    xxx_par_gen.out
c
c
c=============================================================================

      implicit    none
      integer*4   nc1,iodata,iopheno,ioparam,io30,io31,
     +     iseed,ndata,npar,nt
      parameter  (iodata=1,iopheno=23,ioparam=24,io30=30,io31=31,npar=3)
      character   data_file*80,instring*80

      include 'comp.inc'

      np1max = np_max
      nc1max = nc_max
```

```
c     Read inputs

      call getarg(1,instring)
      write(data_file,900) instring

      call getarg(2,instring)
      read(instring,*) iseed

      call getarg(3,instring)
      read(instring,*) nc1

      write(*,*) '  '
      write(*,*) ' Input data file  : ', data_file(1:30)
      write(*,*) ' Iseed            : ', iseed
      write(*,*) ' Nb of components : ', nc1

      nt = nc1 * npar
      nceff = nc1
      npeff = npar

c     First, initialize the random-number generator
      call rninit(iseed)

c     I/O files and units
      call init_files(iodata,iopheno,ioparam,io30,io31,data_file)

c     Read input data
      call read_data(iodata,ndata)
      type *, ' data read in : ', iodata, ndata

c     Set up the control variables for GA
      call set_control_parameters
      type *, ' ctrl OK '

c     Set up the effective arrays, with proper dimensions,
c     and call PIKAIA, print results

      call GA(nt,nc1,npar,ioparam,iopheno,ndata)

c     Close I/O units
      call close_units(ioparam,iopheno,io30,io31)

 900  format(a80)

      stop' *** End of GaGA *** '
      end

c*************************************************************

      subroutine read_data(iodata,n_data)
c     ----------------------------------
c Use: Reads in input data
c
      implicit     none
      integer*4    iodata,i,n_data,nd,nd1
      integer*4    ndata_max
      parameter   (ndata_max=2000)
      real*4       data(ndata_max),sigma(ndata_max)
      real*4       sigma_min, data_max

      common /data/  data,sigma,nd1
      common /props/ sigma_min, data_max, nd

c     Read in the number of data points

      read(iodata,*) n_data

      nd      = n_data
      nd1     = nd

      if (n_data.gt.ndata_max) then
         type *, ' *** Error : n_datam is too small :  ', ndata_max
         type *, ' ***         n_data read in is    :  ', n_data
         type *, ' *** Change value of n_datam in gaga.inc '
```

```
         stop' *** Execution of GaGA stopped *** '
      end if

c     Read in the observed data points and their errors

      read(iodata,*) (data(i),sigma(i),i=1,n_data)

      data_max = -1.0e10
      do i=1, n_data
         data_max = max(data_max,data(i))
      end do

      sigma_min = 0.01          ! in pixel units

c     Close file

      close(iodata)

      return
      end

c*************************************************************

      subroutine init_files(iodata,iopheno,ioparam,io30,io31,data_file)
c     ------------------------------------------------------------------
c Use: Initialises the I/O files
c
      implicit  none
      integer*4 iodata,iopheno,ioparam,io30,io31,io30a,io31a
      character data_file*80,pheno_file*80,param_file*80,
     +     fio30*80,fio31*80

      common /io/ io30a,io31a

      io30a = io30
      io31a = io31

      pheno_file = data_file(1:3)//'_phenotype.out'
      param_file = data_file(1:3)//'_parameter.out'
      fio30      = data_file(1:3)//'_fit_gen.out'
      fio31      = data_file(1:3)//'_par_gen.out'

c     IN  Input data file

      open(unit=iodata,file=data_file,status='old')

c     OUT Final phenotype

      open(unit=iopheno,file=pheno_file,status='unknown')

c     OUT Final parameters

      open(unit=ioparam,file=param_file,status='unknown')

c     Evolution in action :
c
c     OUT Store global fitness of each generation

      open(io30,file=fio30,status='unknown')

c     OUT Store parameters of each generation

      open(io31,file=fio31,status='unknown')

      return
      end

c*************************************************************

      subroutine set_control_parameters
c     ---------------------------------
c Use: Set the control parameters. Utilises the flexibility of
c       PIKAIA
c
```

```
      implicit    none
      integer*4   mc,i
      parameter   (mc=12)
      real*4      ctrl(mc)

      common /control/ ctrl

c     Initialise to default values for security
      do i=1,mc
         ctrl(i) = -1
      end do

c     Number of individuals in a population
      ctrl(1)=50               ! 100 default, 128 maximum

c     Number of generations
         ctrl(2)=500   ! 500 default

c     Number of genes / significant digits in chromosomal encoding
         ctrl(3)=6              ! 32-bit

c     Crossover probability
         ctrl(4) = 0.85         ! must be less than one

c     Mutation mode
c     1    Uniform mutation, constant rate
c     2    Uniform mutation, variable rate based on fitness
c     3    Uniform mutation, variable rate based on distance
c     4    Uniform or creep mutation, constant rate
c     5    Uniform or creep mutation, variable rate based on fitness
c     6    Uniform or creep mutation, variable rate based on distance

         ctrl(5)=5

c     Initial mutation rate
         ctrl(6)=0.005          ! 0.005 default

c     Minimum mutation rate
         ctrl(7)=0.0005         ! >0  (0.0005 default)

c     Maximum mutation rate
         ctrl(8)=0.25           ! <1  (0.25 default)

c     Relative fitness diferential : 0/none  1/maximum (default)
         ctrl(9)=1.

c     Reproduction plan  1 : Full replacement of the generation
c                        2 : Steady-state, replace random
c                        3 : Steady-state, replace worst  (default)
         ctrl(10)=1.

c     Elitism flag       0/1 : off(default)/on [only for ctrl(10)=1 or 2
         ctrl(11)=1.
c
c     Output             0/1/2 : None(default)/Minimal/Verbose
         ctrl(12)=2.

         return
         end

c*************************************************************

      subroutine write_comp_gen(igen,n1,x,io31)
c     ----------------------------------------
c Use: Writes to the 'running' log
c
      implicit   none
      integer*4  n1,i,io31,igen,k,nc1,np1
      real*4     x(n1)
      include    'comp.inc'

      np1 = npeff
      nc1 = nceff
```

```
      call rescale(nc1,np1,x,n1)

      if (igen.ne.0) then
         write(io31,*) igen, nc1, np1 ! generation, Nb components
      else
         write(io31,*) ' Final generation (width in channels)', nc1, np1
      end if

      do i=1,nc1
c         write(io31,910) (comp(i,k),k=1,np1) ! parameters
         write(io31,910) comp(i,1),comp(i,2),1.0/sqrt(comp(i,3))
      end do

 910  format(8(g8.3,2x))
      return
      end

c*************************************************************

      subroutine close_units(ioparam,iopheno,io30,io31)
c     ------------------------------------------------
c Use: Confirms closure of all I/O files
c
      implicit   none
      integer*4  ioparam,iopheno,io30,io31

      close(ioparam)
      close(iopheno)
      close(io30)
      close(io31)

      return
      end

c*************************************************************

      subroutine GA(ntot,nc,npar,ioparam,iopheno,ndata)
c     ------------------------------------------------
c Use: Main subroutine that calls PIKAIA
c
      implicit   none
      integer*4  nc,npar,ntot,istatus,ioparam,iopheno,ndata
      integer*4  i,ntt,ndt
      parameter  (ntt=10*3,ndt=2000)
      real*4     profile(ndt),x(ntt)
      real*4     fitness_function,ctrl(12),f
      external   fitness_function

      common /control/ ctrl

      if (ntot.gt.ntt .or. ndata.gt.ndt) then
         type *, ' Bad dimensions for arrays : ntt  ndt :',ntt,ndt
         type *, ' ntot ndata : ', ntot, ndata
         stop' *** EXIT ***'
      end if

      do i=1,ntot
         x(i) = 0.0
      end do

c     Now call pikaia
      type *, ' calling pik with ntot = ', ntot
      type *, ' ctrl ', ctrl
      call pikaia(fitness_function,ntot,ctrl,x,f,istatus)

c     Print the results
      write(*,*) ' status : ', istatus
      write(*,*) ' params : ', x
      write(*,*) ' fitness: ', f
      write(*,901) ctrl

c     Compute the final phenotype and print the final parameters

      call write_comp_gen(0,ntot,x,ioparam)
```

```
      call get_phenotype(nc,npar,profile,ndata)

      do i=1,ndata
         write(iopheno,*) profile(i)
      end do

 901  format('    ctrl : ',6(f11.6,1x)/10x,6(f11.6,1x))

      return
      end

c=============================================================
c
c     These are added at the compilation stage
c
c     include "fit.f"      ! contains fitness_function
c     include "pikaia.f"   ! contains latest version of pikaia
c
c=============================================================
c*************************************************************
c*************************************************************

      real function fitness_function(n1,x)
c     ------------------------------------
c Use:    Fitness function implemented by Ga-GA routine.
c Input:  n1 -- No. of parameters
c         x  -- genotype array of n1 elements
c
      implicit     none
      integer      n1,i,j,nc1,np1
      real         x(n1),sum
      integer*4    ndata_max, n_data
      parameter   (ndata_max=2000)
      real*4       data(ndata_max),sigma(ndata_max)
      real         profile(ndata_max)
      include      'comp.inc'

      common /data/  data,sigma,n_data

c----------0. initialize parameters and phenotype profile

      nc1 = nceff
      np1 = npeff

      do i=1,nc1
         do j=1,np1
            comp(i,j)= 0.0
         end do
      end do

      do i=1,n_data
         profile(i)=0.
      end do

c----------1. rescale input variables:

      call rescale(nc1,np1,x,n1)

c----------2. compute phenotype :

      call get_phenotype(nc1,np1,profile,n_data)

c----------3. compute reduced chi**2 = {1/[Ndata-(N+1)]} \sum_i [(C-O)/s]_i^2

      sum=0.
      do i=1,n_data
         sum=sum+ ((profile(i)-data(i))/sigma(i))**2.
      end do
      sum=sum/float(n_data-(n1+1))

c----------4. define fitness

      fitness_function=1./sum
```

```
      return
      end

c*************************************************************

      subroutine get_phenotype(nc,n_param,profile,n_data)
c     ---------------------------------------------------
c Use: Takes genotype and computes the phenotype profile
c

      implicit   none
      integer    n_param,n_data,i,j,nc
      real       amp,wid,pos,profile(n_data)
      include    'comp.inc'
      real       genotype
      external   genotype

      do i=1,n_data
         profile(i) = 0.
      end do

      do i=1,nc
         pos = comp(i,1)
         amp = comp(i,2)
         wid = comp(i,3)
         do j=1,n_data
            profile(j)=profile(j)+genotype(pos,amp,wid,j)
         end do
      end do

      return
      end

c*************************************************************

      real function genotype(pos,amp,wid,j)
c     -------------------------------------
c Use: Computes Gaussian phenotype value at channel i
c       for specific genotype

      implicit  none
      integer   j
      real      pos,amp,wid

      genotype = amp*(exp(-wid*((float(j)-pos)**2.)))

      return
      end

c*************************************************************

      subroutine rescale(nc,np,x,n1)
c       ----------------------------
c Use: Rescales each parameter in the genotype for the
c       position, amplitude, & width and store

      implicit  none
      integer   nc,np,nd,n1,i
      real      x(n1)
      real      sigma_min, data_max, cte1
      include   'comp.inc'

      common /props/ sigma_min, data_max, nd

      cte1 = 1.0/(sigma_min*sigma_min)

c     comp(*,np)      nd=N_data
c     np
c     1  pos : 1   - N_data            pos = 1 + x*(N_data-1)
c     2  amp : 0.0 - max(data)         amp = x*max(data)
c     3  wid : 1/sm**2 - 1./N_data**2
c              [wid = 1/sigma**2]  sm = \sigma_min \sim 0.5

      do i=1,nc
```

```
         comp(i,1) = 1.0 + x((np*i)-(np-2))*(float(nd-1))
         comp(i,2) = x((np*i)-(np-1))*(data_max*2.)
         comp(i,3) = (1.0/(x(np*i)*nd))**2
      end do

      return
      end

c*****************************************************************
c*****************************************************************
```

A.2 The Ratio Inversion Technique (RIT) code

```
      program RIT
c     -----------
c*****************************************************************

c=================================================================
c
c    Driver for Ratio Inversion Technique with a Genetic Algorithm
c
c    Scott W McIntosh (V1.04 6 Jun 1998; scott@astro.gla.ac.u
c
c    Usage : RIT <data> <seed> <ngen> <parameters>
c
c     data         line ratios
c     seed         random number generator seed
c     ngen         number of generations
c     parameters   number of discretisation points
c
c=================================================================

      implicit  none
      integer*4   par_max,rcdata,ldata,kdata,rdata
      integer*4   iodata,iopheno
      parameter (par_max=50,rcdata=7,ldata=9)
      parameter (iodata=2,iopheno=23,kdata=3,rdata=5)
      integer   n, iseed, status, ngen, npar
      real      ctrl(12), x(par_max), f, rat_chi
      character data_file*80,instring*80

      external  rat_chi

      common /control/ ctrl

c     Read inputs

      call getarg(1,instring)
      write(data_file,900) instring

      call getarg(2,instring)
      read(instring,*) iseed

      call getarg(3,instring)
      read(instring,*) ngen

      call getarg(4,instring)
      read(instring,*) n

 900  format(a80)

      write(*,*) '  '
      write(*,*) ' Input data file  : ', data_file(1:30)
      write(*,*) ' Iseed            : ', iseed
      write(*,*) ' Generations      : ', ngen
      write(*,*) ' Parameters       : ', n

c     First, initialize the random-number generator
      call start(npar)
      call rninit(iseed)

c     I/O files and units
      call init_files(iodata,iopheno,kdata,rdata,rcdata,ldata,data_file)
```

```
c     Get start-up parameters
      call finit(iodata,kdata,rdata,ldata)

c     Set up the control variables for GA
      call set_control_parameters(ngen,npar)
      type *, ' ctrl OK '
c
      call pikaia(rat_chi,n,ctrl,x,f,status)
c
c     Print the results
      write(*,*) ' status: ',status
      write(*,*) '      x: ',x
      write(*,*) '      f: ',f
      write(*,20) ctrl
 20   format(   '    ctrl: ',6f11.6/10x,6f11.6)

      call output(iopheno,rcdata,n,x)

      end

c**************************************************************

      subroutine set_control_parameters(ngen,npar)
c     --------------------------------
c Use: Set the control parameters. Utilises the flexibility of
c       PIKAIA
c
      implicit    none
      integer*4   mc,i,ngen,npar
      parameter   (mc=12)
      real*4      ctrl(mc)

      common /control/ ctrl

c     Initialise to default values for security
      do i=1,mc
         ctrl(i) = -1
      end do

c     Number of individuals in a population
      ctrl(1)=npar               ! 100 default, 128 maximum

c     Number of generations
         ctrl(2)=ngen   ! 500 default

c     Number of genes / significant digits in chromosomal encoding
         ctrl(3)=6              ! 32-bit

c     Crossover probability
         ctrl(4) = 0.85         ! must be less than one

c     Mutation mode
c     1    Uniform mutation, constant rate
c     2    Uniform mutation, variable rate based on fitness
c     3    Uniform mutation, variable rate based on distance
c     4    Uniform or creep mutation, constant rate
c     5    Uniform or creep mutation, variable rate based on fitness
c     6    Uniform or creep mutation, variable rate based on distance

         ctrl(5)=4

c     Initial mutation rate
         ctrl(6)=0.005          ! 0.005 default

c     Minimum mutation rate
         ctrl(7)=0.0005         ! >0  (0.0005 default)

c     Maximum mutation rate
         ctrl(8)=0.25           ! <1  (0.25 default)

c     Relative fitness diferential : 0/none  1/maximum (default)
         ctrl(9)=1.
```

```
c     Reproduction plan  1 : Full replacement of the generation
c                        2 : Steady-state, replace random
c                        3 : Steady-state, replace worst  (default)
         ctrl(10)=1.

c     Elitism flag       0/1 : off(default)/on [only for ctrl(10)=1 or 2
         ctrl(11)=1.
c
c     Output             0/1/2 : None(default)/Minimal/Verbose
         ctrl(12)=0.

         return
         end

c*************************************************************

      subroutine init_files(iodata,iopheno,kdata,rdata,rcdata,
     + ldata,data_file)
c     ------------------------------------------------------------------
c Use: Initialises the I/O files
c
      implicit  none
      integer*4 iodata,iopheno,kdata,rdata,rcdata,ldata
      character data_file*80,pheno_file*80
      character kern_file*80,r_calc*80,th_err_file*80

      pheno_file = 'fhat.dat'
      r_calc     = 'rat_calc.dat'
      kern_file  = 'kern.dat'
      th_err_file = 'error_dat/th_err.dat'

c     IN  Input data file
      open(unit=iodata,file=data_file,status='unknown')

c     IN  Input kernel file
      open(unit=kdata,file=kern_file,status='unknown')

c     IN  Input th_err file
      open(unit=rdata,file=th_err_file,status='unknown')

c     IN  Input th_err file
      open(unit=ldata,file='ratio_list.dat',status='unknown')

c     OUT Final phenotype
      open(unit=iopheno,file=pheno_file,status='unknown')

c     OUT Recalculated ratios
      open(unit=rcdata,file=r_calc,status='unknown')

      return
      end

c*************************************************************

      subroutine start(npar)
c     ----------------------
c Use: Reads the initialisation file `ratio_start.in'
c
      implicit none
      integer    ndata,npar,smooth,nrat
      double precision lam,scale

      common/misc/ lam,scale,ndata,smooth,nrat

      open(1,file='ratio_start.in',status='old',form='formatted')
      rewind(1)
      read(1,*) ndata,nrat,npar,lam,scale,smooth
      close(1)

      write(*,2)
 2    format(/1x,60('*'),/,
     +     ' *',13x,'Genetic Algorithm Initialisation',13x,'*',/,
     +     1x,60('*'),/)
      write(*,*) 'Number of data points :',ndata
```

```
      write(*,*) 'Number of line ratios :',nrat
      write(*,*) '  Smoothing parameter :',lam
      write(*,*) '    Scaling parameter :',scale

      if (smooth.eq.0)
     +     write(*,*) '  Smoothing Order: Zeroth'
      if (smooth.eq.1)
     +     write(*,*) '  Smoothing Order: First'
      if (smooth.eq.2)
     +     write(*,*) '  Smoothing Order: Second'
      if (smooth.eq.3)
     +     write(*,*) '  Using MaxEnt Smoothing'
      write(*,3)
 3    format(/1x,60('*'),/)

      end

c************************************************************

      subroutine finit(iodata,kdata,rdata,ldata)
c     ------------------------------------------
c Use: Read in all the initial data
c
      implicit none
      integer*4   iodata,kdata,rdata,ldata
      integer   ndata_max,nrat_max,d_max
      parameter (ndata_max=100, nrat_max=100, d_max=50)
      integer   i,j,rl(nrat_max),ndata,smooth,nrat
      double precision r(nrat_max),k(d_max,nrat_max),sigth(nrat_max)
      double precision scale,sigd(nrat_max),lam

      common/data/ r,k,rl,sigth,sigd
      common/misc/ lam,scale,ndata,smooth,nrat

c ----  Read line ratios (R_{obs}) and observational errors
      read(iodata,*) (r(i),sigd(i),i=1,nrat)
      close(iodata)

c ----  Read Kernels
      do 10 i=1,(2*nrat)
         read(kdata,*) (k(j,i),j=1,ndata)
   10 continue
      close(kdata)

c ----  Read theoretical uncertainties
      read(rdata,*) (sigth(i),i=1,nrat)
      close(rdata)

c ----  Read in the ratio pairings (format --> top bottom)
      read(ldata,*) (rl(i),i=1,2*nrat)
      close(ldata)

      do 11 i=1,2*nrat
         rl(i)=rl(i)+1
   11 continue

      return
      end

c************************************************************
      subroutine output(iopheno,rcdata,n,x)
c     -------------------------------------
c Use: Output the final solution and their line
C       ratios (R_{calc})
c
      implicit none
      integer*4   iopheno,rcdata
      integer*4   ndata_max,nrat_max,d_max
      parameter (ndata_max=100, nrat_max=100, d_max=50)
      integer   ndata,i,j,bid,tid,n,nrat,smooth
      integer   rl(nrat_max)
      double precision y(ndata_max),top(ndata_max),bot(ndata_max)
      double precision k(d_max,nrat_max),rc(nrat_max),r(nrat_max)
      double precision lam,scale,sigd(nrat_max),sigth(nrat_max)
```

```
      real      x(n)

      common/data/ r,k,rl,sigth,sigd
      common/misc/ lam,scale,ndata,smooth,nrat

      do 40 i=1,ndata
         write(iopheno,*) (x(i)*scale)
   40 continue

      do 3 i=1,ndata
         y(i)=x(i)*scale
    3 continue

      do i=1,nrat
         top(i)=0.d0
         bot(i)=0.d0
      enddo

      do 4 i=1,nrat
         tid=rl((2*i)-1)
         bid=rl(2*i)
         do 5 j=1,ndata
            top(i)=top(i) + k(j,tid)*y(j)
            bot(i)=bot(i) + k(j,bid)*y(j)
    5    continue
         rc(i)=top(i)/bot(i)
    4 continue

      do 7 i=1,nrat
         write(rcdata,*) rc(i)
    7 continue

      return
      end

c=============================================================
c
c     These are added at the compilation stage
c
c     include "rat_chi.f" ! contains fitness function
c     include "pikaia.f"  ! contains latest version of pikaia
c
c=============================================================
c*************************************************************
c*************************************************************

      real function rat_chi(n,x)
c     --------------------------

c=============================================================
c         Fitness function for Ratio Inversion
c     Scott McIntosh (scott@astro.gla.ac.uk) 11/5/98
c
c Use: Computes \chi^{2} estimator for RIT
c
c Input:  n -- No. of parameters (discretisation points)
c         x  -- genotype array of n elements
c
c=============================================================

      implicit none
      integer      n,i,j,ndata,tid,bid
      integer      ndata_max,nrat_max,d_max
      parameter    (ndata_max=100, nrat_max=100, d_max=50)
      integer      rl(nrat_max),smooth,nrat
      real         x(n),a1,b1,a2,b2,rat_chi
      double precision h,lam,totd,ytot,yav,sum,scale
      double precision top(ndata_max),bot(ndata_max),sumn(ndata_max)
      double precision deriv(ndata_max),y(ndata_max),sigd(nrat_max)
      double precision k(d_max,nrat_max),r(nrat_max),sigth(nrat_max)

      common/data/ r,k,rl,sigth,sigd
      common/misc/ lam,scale,ndata,smooth,nrat
```

```
c----------1. Initialise and rescale variables:

      ytot=0.d0
      do 2 i=1,ndata
         y(i)=x(i)*scale
         deriv(i)=0.
         ytot=ytot+y(i)
    2 continue

      do i=1,nrat
         top(i)=0.
         bot(i)=0.
         sumn(i)=0.
      enddo

      yav=ytot/ndata

c----------2. Compute smoothness of solution

      sum=0.
      h=6.35
      totd=0.

c----------2a Parabola fitting

      a1=(y(1)-y(3))-2*(y(1)-y(2))/(2*(h*h))
      b1=((y(1)-y(2))/h) - a1*2*h
      a2=(y(ndata)-y(ndata-2))-2*(y(ndata)-y(ndata-1))/(2*(h*h))
      b2=((y(ndata)-y(ndata-1))/h) - a2*2*h

      if (smooth.eq.0) then
         do i=1,ndata
            deriv(i) = y(i)
         end do
      endif

      if (smooth.eq.1) then
         do i=1,(ndata-3)
            deriv(i) = ((-y(i+3) +4*y(i+1) -3*y(i))/(2*h))
         end do
         do i=(ndata-2),(ndata-1)
            deriv(i) = ((y(i+1)-y(i))/(h))
         end do
         deriv(ndata) = a2*(h)+b2
      endif

      if (smooth.eq.2) then
         do i=1,(ndata-4)
            deriv(i) = ((-y(i+3)+4*y(i+2)-5*y(i+1)+2*y(i))/(h*h))
         end do
         do i=(ndata-3),(ndata-2)
            deriv(i) = ((y(i+2)-2*y(i+1)+y(i))/(h*h))
         end do
         deriv(ndata-1)=((y(ndata)-2*y(ndata-1)+y(ndata-2))/(h*h))
         deriv(ndata)=2*a2
      endif

c----------2b. Integrate to obtain modulus

      do i=1,ndata
         totd=totd+(deriv(i))**2
      enddo
      totd=sqrt(totd)

c----------2c. MaxEnt smoothing functional

      if (smooth.eq.3) then
         do i=1,ndata
            totd = totd + (y(i)/yav)*log(y(i)/yav)
         end do
      endif

c----------3. Compute X(R_{obs},R_{calc})=(R_{calc} - R_{obs})^2
```

```
      do 4 i=1,nrat
         tid=rl((2*i)-1)
         bid=rl(2*i)
         do 1 j=1,ndata
            top(i) = top(i) + (k(j,tid)*y(j))
            bot(i) = bot(i) + (k(j,bid)*y(j))
    1    continue
    4 continue

      do 7 i=1,nrat
         sumn(i) = (r(i)-(top(i)/bot(i)))**2
    7 continue

      do 5 i=1,nrat
         sum = sum + (sumn(i)/(sigth(i)**2 + sigd(i)**2))
    5 continue

c For MaxEnt calculation
      if (smooth.eq.3) then
         sum = sum + (lam*totd)
         rat_chi = 1./sum
c For derivative calculation
      else
         sum = sum + (lam*totd)
         rat_chi=1./(1.+sum)
      endif

      return
      end

c*************************************************************
```

Appendix B

Some SELECTOR details

In Chapter 5 we discussed the application of the SELECTOR genetic algorithm to the optimisation of the condition number of the Differential Emission Measure (DEM) inverse problems (and hence improve the numerical stability of the inferred solution to data noise). We performed this by allowing SELECTOR to search for an optimal set of emission lines in the SOHO CDS/SUMER wavelength range (150-1610 Å). In the following sections we discuss the condition number estimator (primarily to increase the speed of the algorithm) of Cline et al. (1979) and provide the Fortran-77 code of the algorithm.

B.1 Condition number estimation

In Section 5.1, while introducing the mechanics of the SELECTOR GA, we drew the reader's attention to the fact that we are able to calculate the condition number C_K of an $m \times n$ matrix K using either a full Singular Value Decomposition (SVD; see Section 2.1.3.2) or by using an *estimate* for C_K discussed by Cline et al. (1979). The former is known to be computationally expensive $O(n^3)$ (n being the major dimension of matrix K) whereas the latter is only $O(n^2)$. The benefits of implementing the condition number estimate in SELECTOR are clear when considering the number of calculations required in a single evolutionary run of 5,000 generations, say. With 100 individuals in the population and $n(=m) = 30$ we have, for one run,

$$5000 \times 100 \times 30^2 = 4.5 \times 10^8$$

calculations, excluding genetic operations.

Recalling the discussion of Section 2.1.2, we see that for the linear system $\hat{\mathbf{g}} = K\hat{\mathbf{f}}$ where the data ($\hat{\mathbf{g}} = \mathbf{g} + \delta\mathbf{g}$) and solution ($\hat{\mathbf{f}} = \mathbf{f} + \delta\mathbf{f}$), with their respective errors ($\delta\mathbf{g}$ and $\delta\mathbf{f}$),

experience error amplification of the order, when matrix K is poorly conditioned

$$\frac{\|\delta \mathbf{f}\|_p}{\|\mathbf{f}\|_p} \leq \|K\|_p \, \|K^{-1}\|_p \, \frac{\|\delta \mathbf{g}\|_p}{\|\mathbf{g}\|_p} \tag{B.1}$$

where $\|\cdot\|_p$ is a norm (with $p = 1, 2, \infty$; see Section 2.1.2). We also note that the condition number C_K of the system is defined as (cf. equation (2.19))

$$C_K = \|K\|_p \, \|K^{-1}\|_p \, = \frac{\sigma_{max}}{\sigma_{min}} \tag{B.2}$$

with the second equality holding when $p = 2$ for σ_{max} and σ_{min}, the maximum and minimum singular values of K respectively. However, the numerically fast estimation of C_K without SVD is not straightforward and hinges upon the calculation of $\|K^{-1}\|$ **without** computing the inverse matrix K^{-1}.

So, following the discussion of Cline et al. (1979) we consider the LU decompositon (see, e.g., Press et al. 1992) of a matrix A

$$PAQ = LU \tag{B.3}$$

where L, U, P and Q are unit lower-triangular, upper-triangular and pivoting (from Gaussian Elimination with Q normally equal to the identity matrix, see Sneddon 1972) matrices respectively and we must estimate $\|A^{-1}\|$ by solving the hypothetical linear system $A\mathbf{x} = \mathbf{b}$ where we have complete freedom to choose the right-hand side $\mathbf{b}$ subject to $\mathbf{x}$ being "big enough" that

$$\max \frac{\|\mathbf{x}\|}{\|\mathbf{b}\|} \approx \sigma_{min} = \|U^{-1}\|_2 \tag{B.4}$$

holds.

For simplicity, in the code, we write $PAQ = A$ and we seek the ratio $\|\mathbf{y}\|_\infty / \|\mathbf{x}\|_\infty$ where the vector $\mathbf{y}$ is defined by :

$$LU\mathbf{y} = \mathbf{x} \, , \; (LU)^T \mathbf{x} = \mathbf{b} \, . \tag{B.5}$$

The first step is to find the solution of $(LU)^T \mathbf{x} = \mathbf{b}$ which is performed in two stages by finding the solution to

$$U^T \mathbf{z} = \mathbf{b} \text{ and } \mathrm{L}^{\mathrm{T}} \mathbf{x} = \mathbf{z} \tag{B.6}$$

for which we wish to maximise the solution to $U^T \mathbf{z} = \mathbf{b}$ subject to the constraint that $\mathbf{z}$ is "large" relative to $\mathbf{b}$. This is done by choosing the k^{th} element of $\mathbf{b}$ to belong to the set $\{-1, +1\}$. Algorithmically, for step k, we have (for *any* triangular matrix T - U^T in this case) :

$$z(k)^+ = (1 - p(k))/T(k,k)$$

$$s(k)^+ = |z(k)^+| + \|p(1:k-1) + T(1:k-1,k)z(k)^+\|_2$$

$$z(k)^- = (-1 - p(k))/T(k,k)$$

$$s(k)^- = |z(k)^-| + \|p(1:k-1) + T(1:k-1,k)z(k)^-\|_2$$

where $z(k)^+$, $z(k)^-$ are upper and lower estimates of element $z(k)$, $s(k)^+$ and $s(k)^-$ are their respective running sums and the value $p(k)$ determines the sign of $b(k)$ ($p(k) \geq 0$ sets $b(k) = -1$ and if $p(k) < 0$ sets $b(k) = -1$). Furthermore $z(k)$ is set to $z(k)^+$ if $s(k)^+ \geq s(k)^-$ and to $z(k)^-$ otherwise. Repeating for all k we obtain a 'large' estimate for $\mathbf{z}$. The full subroutine is called `strco.f` and can be found in Golub & Van Loan (1989) and in the following section.

The condition number estimate is then readily achieved by solving the triangular systems $L^T\mathbf{x} = \mathbf{z}$ (giving $\mathbf{x}$), $L\mathbf{w} = P\mathbf{x}$ (giving $\mathbf{w}$; a new intermediate vector) and $U\mathbf{y} = \mathbf{w}$ finally yeilding $\mathbf{y}$, by back-substitution (Sneddon 1972). The estimate of C_K is then readily obtained by calculating the ∞-norm ($p = \infty$) of $\mathbf{x}$ and $\mathbf{y}$ and taking their ratio.

B.2 The SELECTOR code

```
      Program selector
c     ----------------
c=====================================================================
c
c   Genetic Algorithm to seek the optimal subset of emission
c lines from the set formed by the set of lines in the SOHO
c CDS/SUMER wavelength range (150 -- 1610 Angstroms).
c
c   The optimal set is selected via the condition number of
c the kernel matrix that constitutes the phenotype. To ensure
c that the 'genes' (parameters; line indices) are not repeated in
c the gentype at any stage of the process we implement Ranked
c Ordinal Representation (ROR; see Michalewicz 1994).
c
c   The condition number can be estimated in two ways through the
c 'calc' variable in the input file. If calc eq. 1 then SELECTOR
c will perform the calculation using a full n^{3} SVD
c (Press et al. 1992) analysis else it will use the n^{2} condition
c number estimate (involving an LU decomposition) from
c Cline et al. (1979) and discussed in Golub & Van Loan (1990). The
c latter is a good order of magnitude estimate and is significantly
c faster.
c
c     Scott W McIntosh (V2.1 6 Dec 1997; scott@astro.gla.ac.uk)
c
c=====================================================================
      Implicit none

c Constants
      integer    NMAX,PMAX,DMAX
      parameter (NMAX = 50,PMAX = 100,DMAX=100)

c Variables
      integer    seed,maxlines,npar,n,nchoice,ngen,irep,new
      integer    ip,ip1,ip2,kdim,i,newtot,ielite,calc,ig
```

```
      real       fdif,pmut,pcross

c Arrays
      integer    ifit(PMAX),jfit(PMAX)
      integer    ph(NMAX,2),oldph(NMAX,PMAX),newph(NMAX,PMAX)
      integer    gn1(NMAX),gn2(NMAX)
      real       fitns(PMAX),work(NMAX,DMAX),space(NMAX)

c Functions
      real     urand
      external urand

c     Read in the full line data into common block
         Call setup(n,npar,maxlines,ngen,nchoice,pcross,
     +        pmut,fdif,kdim,ielite,seed,irep,out)
         Call rninit(seed)

c     Create the initial population
         Call initpop(NMAX,n,npar,init,maxlines,oldph)

         Call eval_fit(NMAX,n,npar,fitns,oldph,kdim,work,
     +        space)
         wst=1.e0

c     Rank initial population
         Call rnkpop(npar,fitns,ifit,jfit)

c     Main program loop (Generation)
         do 10 ig=1,ngen
            newtot=0

c     Main Population Loop
            do 20 ip=1,npar/2

c     Breed population (2 parents at a time)
c        1. select two parents
               Call select(npar,jfit,fdif,ip1)
   21          Call select(npar,jfit,fdif,ip2)
               if (ip1.eq.ip2) goto 21

c    transform ph to gh
               do 101 i=1,n
                  gn1(i)=oldph(i,ip1)
                  gn2(i)=oldph(i,ip2)
  101          continue

c        2. breed (watching for uniqueness)
               Call cross(n,pcross,gn1,gn2)
               Call mutate(n,maxlines,pmut,gn1)
               Call mutate(n,maxlines,pmut,gn2)

c   transform gh to ph
               do 102 i=1,n
                  ph(i,1)=gn1(i)
                  ph(i,2)=gn2(i)
  102          continue

c        3. insert into population
            if (irep.eq.1) then
               Call genrep(NMAX,n,npar,ip,ph,newph)
            else
               Call stdrep(NMAX,n,npar,irep,ielite,
     +                     ph,oldph,fitns,ifit,jfit,new)
               newtot = newtot+new
            endif

   20       continue

c     Evaluate new generation + Rank + Order Population
            Call newpop(NMAX,n,npar,oldph,newph,
     +           ifit,jfit,fitns,newtot,kdim,maxlines,ielite)

c     Report on progress after every X generations
c     and handle end event !!
```

```
            Call report(NMAX,n,npar,fitns,oldph,ig,ngen,
     +           kdim,ifit,out,wst,worst)
c     End of Main loop (Generation)
   10    continue

         close(4)

      end

c*********************************************************************

      subroutine setup(n,npar,maxlines,ngen,nchoice,pcross,
     +               pmut,fdif,kdim,ielite,calc,seed,irep)
c     ------------------------------------------------------
c=====================================================================
c    Performs reading of initialization data
c          For use with Selector
c=====================================================================
c Common block details for fitness evaluation
      common/data/ kern(100,200)
      real         kern
c Variables
      integer      nchoice,maxlines,n,npar,maxlines
      integer      kdim,ngen,i,ielite,calc,j,seed
      real         pmut,pcross,fdif

c     Read in data, line list and probably config data

      open(3,file='select.init')
      rewind(3)
      read(3,*) n,npar,maxlines,ngen,nchoice,kdim,pcross,
     +          pmut,fdif,ielite,calc,seed,irep
      close(3)
c     print header and info
      write(*,2) ngen,npar,n,maxlines,nchoice,kdim,pcross,
     +          pmut,fdif,calc,ielite,seed
    2 format(/1x,60('*'),/,
     +      ' *',13x,'SELECTOR Genetic Algorithm Report',12x,'*',/,
     +           1x,60('*'),//,
     +      '   Number of Generations evolving: ',i4,/,
     +      '       Individuals per generation: ',i4,/,
     +      '    Number of Chromosome segments: ',i4,/,
     +      '    Number of Lines to chose from: ',i4,/,
     +      '                Number of choices: ',i4,/,
     +      '              Dimension on Matrix: ',i4,/,
     +      '            Crossover probability: ',f9.4,/,
     +      '            Initial mutation rate: ',f9.4,/,
     +      '    Relative fitness differential: ',f9.4,/,
     +      '       Full SVD(1) or Estimate(0): ',i4,/,
     +      '           Elitism - yes(1) no(0): ',i4,/,
     +      '                             Seed: ',i8)
      if (irep.eq.1) write(*,4) 'Full generational replacement'
      if (irep.eq.2) write(*,4) 'Steady-state-replace-random'
      if (irep.eq.3) write(*,4) 'Steady-state-replace-worst'
    4 format(
     +      '                Reproduction Plan: ',A)

      open(2,file='file.dat')
      rewind(2)
      read(2,*) ((kern(i,j),j=1,kdim),i=1,maxlines)
      close(2)
      write(*,1005)
      write(*,1006)
 1005 format(/1x,60('*'),/)
 1006 format(1x,'Gen.',3x,'Best',10x,'Median')

      return
      end

c*********************************************************************

      subroutine report(ndim,n,np,fit,pop,ngen,mg,kdim,ifit)
c     ------------------------------------------------------
```

```
c=====================================================================
c Performs output on the end of each generation
c          For use with Selector
c=====================================================================

c Common block details for fitness evaluation
      common/data/ kern(100,200)
      real         kern
c Variables
      integer      i,j,n,np,ndim,mg,kdim,ngen
c Arrays
      integer      pop(ndim,np),ifit(np)
      real         fit(np),best(100,100)

      open(4,file='converge.log')

c Write best and median child  to screen
c Write best child to log (fitness + ids)
      if (mod(ngen,10).eq.0) then
         write(*,*) ngen,fit(ifit(np)),fit(ifit(np/2))
         do 1 i=1,n
            write(*,1003) pop(i,ifit(np)),pop(i,ifit(np/2))
    1    continue
      endif
      write(4,*) fit(ifit(np))

 1001 format(/1x,i4,g9.7,5x,g9.7)
 1002 format(g10.7)
 1003 format(6x,i4,10x,i4)

c Write best Matrix to file on completion of mg generations
      if (ngen.eq.mg) then
         open(5,file='best.dat')
         open(6,file='sel_out.dat')
         rewind(5)
         rewind(6)

         do 5 i=1,n
            write(6,*) pop(i,ifit(np))
            do 6 j=1,kdim
               best(i,j)=kern(pop(i,ifit(np)),j)
    6       continue
    5    continue
         do 7 i=1,n
            write(5,*) (best(i,j),j=1,kdim)
    7    continue
         close(5)
         close(6)
      end if

      end

c**********************************************************************

      subroutine initpop(ndim,n,npar,pop,maxlines)
c     ----------------------------------------
c=====================================================================
c    Calculates initial population of integers
c          For use with Selector
c=====================================================================
      integer  n,npar,maxlines,pop(ndim,npar)
      integer  ip,j,ndim,oldph(ndim,npar)
      integer  ip,j,ndim,temp(100),p_temp(100)
      real     urand
      external urand

      do 1 ip=1,npar
         do 2 j=1,n
            pop(j,ip) = int(maxlines*urand())+1
            temp(j) = pop(j,ip)
    2    continue
         Call decoder(maxlines,n,temp,p_temp)
         do 3 j=1,n
            oldph(j,ip) = p_temp(j)
```

```
    3    continue
    1 continue

      end

c**********************************************************************

      subroutine select(np,jfit,fdif,idad)
c     ----------------------------------
c=====================================================================
c    Selects parents from population, using a roulette wheel
c   algorithm with relative fitness of phenotypes as 'hit'
c   probabilities [Davis 91 Ch.1]. For use with Selector
c=====================================================================
      implicit none
      integer        np, jfit(np), idad, np1, i
      real           fdif, dice, rtfit, urand
      external       urand

      np1 = np+1
      dice = urand()*np*np1
      rtfit = 0.
      do 1 i=1,np
         rtfit = rtfit+np1+fdif*(np1-2*jfit(i))
         if (rtfit.ge.dice) then
            idad=i
            goto 2
         endif
    1 continue
c     Assert: loop will never exit by falling through

    2 return
      end

c**********************************************************************

      subroutine cross(n,pcross,gn1,gn2)
c     --------------------------------
c=====================================================================
c  Breeds two parent chromosomes into two offspring
c    chromosomes: machanism is simple cross-over at
c    the position ispl. For use with Selector
c=====================================================================

      implicit none
c
      integer        n, i, ispl, t, gn1(n), gn2(n)
      real           pcross, urand
      external       urand

c     Use crossover probability to decide whether a crossover occurs
      if (urand().lt.pcross) then
c        Compute crossover point
         ispl=int(urand()*(n-1))+1
c        Swap genes at ispl and above
         do 10 i=ispl,n
            t=gn2(i)
            gn2(i)=gn1(i)
            gn1(i)=t
   10    continue
      endif

      return
      end

c**********************************************************************

      subroutine newpop(ndim,n,np,oldph,newph,ifit,jfit,
     +                  fitns,nnew,kdim,maxlines,ielite,calc)
c     ----------------------------------------------------
c=====================================================================
c    Replaces old population by new, also recomputes fitnesses
c       and ranks of the new population
c    For use with Selector (c) Scott McIntosh
```

```
c=====================================================================

      implicit none
      integer    NMAX,PMAX,ndim, np, n, maxlines, ielite, calc
      integer    oldph(ndim,np), newph(ndim,np), i, k
      integer    ifit(np), jfit(np), nnew, kdim
      parameter  (NMAX = 50,PMAX = 100)
      real           fitns(np),work(NMAX,PMAX),space(PMAX),st(PMAX)
      real       ff
      external   ff

c     if using elitism, introduce in new population fittest of old
c     population (if greater than fitness of the individual it is
c     to replace)

      if (ielite.eq.1 .and. ff(n,newph(1,1),calc).lt.fitns(ifit(np))) then
         do 1 k=1,n
            newph(k,1)=oldph(k,ifit(np))
    1    continue
         nnew = nnew-1
      endif

c     replace population
      do 2 i=1,np
         do 3 k=1,n
            oldph(k,i)=newph(k,i)
    3    continue
         fitns(i)=ff(n,oldph(1,i),calc)
    2 continue

c     compute new population fitness rank order
      Call rnkpop(np,fitns,ifit,jfit)

      return
      end

c**********************************************************************

      subroutine eval_fit(ndim,n,npar,fitns,family,kdim,
     +                    work,space,calc)
c     ------------------------------------------------
c=====================================================================
c    Calculates Fitness of population array in this
c        case, it is the condition number
c=====================================================================
      implicit none
c Common block details for fitness evaluation
      common/data/ kern(100,200)
      real         kern
c Variables
      integer      n,npar,i,j,kdim,m,ndim,calc
      real         cond,info,info2,wmin,wmax
c Arrays
      integer      index(100),family(ndim,npar)
      real         work(n,kdim),space(n),fitns(npar)
      real         v(100,100)
c Uses
c strco and ludcmp routines

c Compute working space matrix
      do 4 i=1,npar
         do 5 m=1,n
            do 6 j=1,kdim
               work(m,j)=kern(family(m,i),j)
    6       continue
    5    continue

c Is it full SVD or estimator ?
         if (calc.eq.1) then
             Call svdcmp(work,n,kdim,n,kdim,space,v)
             wmax=space(1)
             wmin=space(1)
             do 7 m=1,n
                if(space(m).gt.wmax)then
```

```
                  wmax=space(m)
               endif
               if((space(m).lt.wmax).and.(space(m).gt.0.d0))then
                  wmin=space(m)
               endif
    7        continue
             cond=(wmin/wmax)
        else
c Compute LU decomposition
            info=0.d0
            Call ludcmp(work,n,kdim,index,info)
c Compute estimate of condition number for workspace matrix
            info2=0.d0
            Call strco(work,n,kdim,cond,space,info2)
        endif

        fitns(i)=cond
    4 continue

      end

c**********************************************************************

      subroutine rnkpop(n,arrin,indx,rank)
c     ------------------------------------
c=====================================================================
c    Uses RQSort to sort population fitness levels in
c  array rank (ascending order). For use with Selector
c=====================================================================
      implicit none
      integer    n,indx(n),rank(n),i
      real       arrin(n)
      external rqsort

c     Compute the key index
      Call rqsort(n,arrin,indx)
c     ...and the rank order

      do 1 i=1,n
         rank(indx(i)) = (n-i)+1
    1 continue
      return
      end

c**********************************************************************

      subroutine genrep(ndim,n,np,ip,ph,newph)
c     ----------------------------------------
c=====================================================================
c     Full generation Replacement subroutine
c     For use with Selector
c=====================================================================
      implicit none
      integer ndim,n,np,ip,i1,i2,k
      integer ph(ndim,2),newph(ndim,np)

      i1 = 2*ip - 1
      i2 = i1 + 1
      do 1 k=1,n
         newph(k,i1) = ph(k,1)
         newph(k,i2) = ph(k,2)
    1 continue

      return
      end

c**********************************************************************

      subroutine decoder(m,n,s,lnlist)
c     --------------------------------
c=====================================================================
c     Converts an ordinal vector (integer) into a line list (integer)
c     using ROR;
c
```

```
c     INPUT:
c     m:      scalar integer, size of line sample
c     n:      scalar integer, size of line subset being evolved
c     s:      integer, size n, ordinal vector defining selection
c
c     OUTPUT:
c     lnlist: integer, size n, line selection
c=======================================================================
      implicit none
      integer NMAX,MMAX
      parameter(NMAX=40,MMAX=133)
c Input
      integer n,m
      integer s(n)
c Output
      integer lnlist(n)
c Local
      integer indx(NMAX),sample(MMAX)
      external indexx
      integer i,j,ii,jj

c     1. construct sample vector
      do i=1,m
 sample(i)=i
      enddo

c     2. rank selection vector
      Call indexx(n,s,indx)

c     3. build line list
      do i=1,n
 ii=indx(n-i+1)
 jj=s(ii)
 if(jj.gt.(m-i+1)) jj=jj-(m-i+1)
 lnlist(i)=sample( jj )
 do j=jj,m-1
    sample(j)=sample(j+1)
 enddo
      enddo

      return
      end

c**********************************************************************

      SUBROUTINE indexx(n,arr,indx)
c     ----------------------------
c=======================================================================
c     Ranks the n-element array (arr; See Press el al.1992)
c=======================================================================
      INTEGER n,indx(n),M,NSTACK
      REAL arr(n)
      PARAMETER (M=11,NSTACK=50)
      INTEGER i,indxt,ir,itemp,j,jstack,k,l,istack(NSTACK)
      REAL a
      do 11 j=1,n
        indx(j)=j
11    continue
      jstack=0
      l=1
      ir=n
1     if(ir-l.lt.M)then
        do 13 j=l+1,ir
          indxt=indx(j)
          a=arr(indxt)
          do 12 i=j-1,l,-1
            if(arr(indx(i)).le.a)goto 2
            indx(i+1)=indx(i)
12        continue
          i=l-1
2         indx(i+1)=indxt
13      continue
        if(jstack.eq.0)return
        ir=istack(jstack)
```

```
        l=istack(jstack-1)
        jstack=jstack-2
      else
        k=(l+ir)/2
        itemp=indx(k)
        indx(k)=indx(l+1)
        indx(l+1)=itemp
        if(arr(indx(l+1)).gt.arr(indx(ir)))then
          itemp=indx(l+1)
          indx(l+1)=indx(ir)
          indx(ir)=itemp
        endif
        if(arr(indx(l)).gt.arr(indx(ir)))then
          itemp=indx(l)
          indx(l)=indx(ir)
          indx(ir)=itemp
        endif
        if(arr(indx(l+1)).gt.arr(indx(l)))then
          itemp=indx(l+1)
          indx(l+1)=indx(l)
          indx(l)=itemp
        endif
        i=l+1
        j=ir
        indxt=indx(l)
        a=arr(indxt)
3       continue
          i=i+1
        if(arr(indx(i)).lt.a)goto 3
4       continue
          j=j-1
        if(arr(indx(j)).gt.a)goto 4
        if(j.lt.i)goto 5
        itemp=indx(i)
        indx(i)=indx(j)
        indx(j)=itemp
        goto 3
5       indx(l)=indx(j)
        indx(j)=indxt
        jstack=jstack+2
        if(jstack.gt.NSTACK)pause 'NSTACK too small in indexx'
        if(ir-i+1.ge.j-l)then
          istack(jstack)=ir
          istack(jstack-1)=i
          ir=j-1
        else
          istack(jstack)=j-1
          istack(jstack-1)=l
          l=i
        endif
      endif
      goto 1
      END
c  (C) Copr. 1986-92 Numerical Recipes Software

c**********************************************************************

      subroutine stdrep
     +   (ndim,n,np,irep,ielite,ph,oldph,fitns,ifit,jfit,nnew)
c     ------------------------------------------------------
c=======================================================================
c     steady-state reproduction: insert offspring pair into population
c     only if they are fit enough (replace-random if irep=2 or
c     replace-worst if irep=3).
c=======================================================================
      implicit none
      integer        ndim, n, np, irep, ielite, ifit(np)
      integer        jfit(np), nnew, i, j, k, i1, if1
      real           ff, ph(ndim,2), oldph(ndim,np), fitns(np), fit
      real           urand
      external       ff, urand

      nnew = 0
      do 1 j=1,2
```

```
c        1. compute offspring fitness (with caller's fitness function)
         fit=ff(n,ph(1,j),calc)

c        2. if fit enough, insert in population
         do 20 i=np,1,-1
            if (fit.gt.fitns(ifit(i))) then

c              make sure the phenotype is not already in the population
               if (i.lt.np) then
                  do 5 k=1,n
                     if (oldph(k,ifit(i+1)).ne.ph(k,j)) goto 6
    5             continue
                  goto 1
    6             continue
               endif

c              offspring is fit enough for insertion, and is unique
c              (i) insert phenotype at appropriate place in population
               if (irep.eq.3) then
                  i1=1
               else if (ielite.eq.0 .or. i.eq.np) then
                  i1=int(urand()*np)+1
               else
                  i1=int(urand()*(np-1))+1
               endif
               if1 = ifit(i1)
               fitns(if1)=fit
               do 21 k=1,n
                  oldph(k,if1)=ph(k,j)
   21          continue

c              (ii) shift and update ranking arrays
               if (i.lt.i1) then

c                 shift up
                  jfit(if1)=np-i
                  do 22 k=i1-1,i+1,-1
                     jfit(ifit(k))=jfit(ifit(k))-1
                     ifit(k+1)=ifit(k)
   22             continue
                  ifit(i+1)=if1
               else

c                 shift down
                  jfit(if1)=np-i+1
                  do 23 k=i1+1,i
                     jfit(ifit(k))=jfit(ifit(k))+1
                     ifit(k-1)=ifit(k)
   23             continue
                  ifit(i)=if1
               endif
               nnew = nnew+1
               goto 1
            endif
   20    continue

    1 continue

      return
      end

c**********************************************************************

      subroutine mutate(n,max,pmut,gn)
c     ------------------------------
c=====================================================================
c  Performs single gene mutation if conditions allow.
c=====================================================================
      implicit none
      integer n,gn(n),i,max
      real pmut, urand
      external urand
```

```
      do 10 i=1,n
         if (urand().lt.pmut) then
                gn(i)=int(urand()*max) + 1
         end if
   10 continue

      return
      end

c*********************************************************************

      Function ff(n,x,job)
c     ----------------
c=====================================================================
c The fitness evaluation function
c=====================================================================
      common/data/ kern(100,200)

      real         kern, cond, ff, info, info2, work(30,30), space(30)
      integer      n, i, j, kdim, m, index(30), x(n), job, indx(n)
      external     rqsort

      kdim=n
c     Compute working space matrix
      do 5 m=1,n
         do 6 j=1,kdim
            work(m,j)=kern(x(m),j)
    6    continue
    5 continue

c Is it full SVD or estimator ?
         if (calc.eq.1) then
             Call svdcmp(work,n,kdim,n,kdim,space,v)
             Call indexx(n,n,indx)
             cond=(space(indx(n))/space(indx(0)))
        else
c Compute LU decomposition
            info = 0.d0
            Call ludcmp(work,n,kdim,index,info)
c Compute estimate of condition number for workspace matrix
            info2 = 0.d0
            Call strco(work,n,kdim,cond,space,info2)
        endif

      ff=cond
      return
      end

c*********************************************************************

      subroutine ludcmp(a,n,np,indx,d)
c     --------------------------------
c=====================================================================
c  Performs LU decomposition of Matrix A (See Press el al.1992)
c=====================================================================
      integer   n, np, indx(n), nmax
      integer   i, imax, j, k
      real      d,a(np,np),tiny
      real      aamax,dum,sum,vv(nmax)
      parameter (nmax=500,tiny=1.0e-20)

      d=1.
      do 12 i=1,n
        aamax=0.
        do 11 j=1,n
          if (abs(a(i,j)).gt.aamax) aamax=abs(a(i,j))
11      continue
        if (aamax.eq.0.) pause 'singular matrix in ludcmp'
        vv(i)=1./aamax
12    continue
      do 19 j=1,n
        do 14 i=1,j-1
          sum=a(i,j)
          do 13 k=1,i-1
```

```
            sum=sum-a(i,k)*a(k,j)
13        continue
          a(i,j)=sum
14      continue
        aamax=0.
        do 16 i=j,n
          sum=a(i,j)
          do 15 k=1,j-1
            sum=sum-a(i,k)*a(k,j)
15        continue
          a(i,j)=sum
          dum=vv(i)*abs(sum)
          if (dum.ge.aamax) then
            imax=i
            aamax=dum
          endif
16      continue
        if (j.ne.imax)then
          do 17 k=1,n
            dum=a(imax,k)
            a(imax,k)=a(j,k)
            a(j,k)=dum
17        continue
          d=-d
          vv(imax)=vv(j)
        endif
        indx(j)=imax
        if(a(j,j).eq.0.)a(j,j)=tiny
        if(j.ne.n)then
          dum=1./a(j,j)
          do 18 i=j+1,n
            a(i,j)=a(i,j)*dum
18        continue
        endif
19    continue
      return
      end
c  (c) copr. 1986-92 numerical recipes software

c**********************************************************************

      subroutine strco(t,ldt,n,rcond,z,job)
c     ------------------------------------
c=====================================================================
c    keywords  condition,factor,linear algebra,linpack,matrix,triangular
c    author  moler, c. b., (u. of new mexico)
c    purpose  estimates the condition of a real triangular matrix.
c
c     strco estimates the condition of a real triangular matrix.
c
c     on entry
c
c        t       real(ldt,n)
c                t contains the triangular matrix.  the zero
c                elements of the matrix are not referenced, and
c                the corresponding elements of the array can be
c                used to store other information.
c
c        ldt     integer
c                ldt is the leading dimension of the array t.
c
c        n       integer
c                n is the order of the system.
c
c        job     integer
c                = 0         t  is lower triangular.
c                = nonzero   t  is upper triangular.
c
c     on return
c
c        rcond   real
c                an estimate of the reciprocal condition of  t .
c                for the system  t*x = b , relative perturbations
c                in  t  and  b  of size  epsilon  may cause
```

```
c                relative perturbations in  x  of size  epsilon/rcond .
c                if  rcond  is so small that the logical expression
c                          1.0 + rcond .eq. 1.0
c                is true, then  t  may be singular to working
c                precision.  in particular,  rcond  is zero  if
c                exact singularity is detected or the estimate
c                underflows.
c
c        z       real(n)
c                a work vector whose contents are usually unimportant.
c                if  t  is close to a singular matrix, then  z  is
c                an approximate null vector in the sense that
c                norm(a*z) = rcond*norm(a)*norm(z) .
c
c***references  dongarra j.j., bunch j.r., moler c.b., stewart g.w.,
c                *linpack users  guide*, siam, 1979.
c======================================================================
      integer ldt,n,job
      real t(ldt,1),z(1)
      real rcond
c
      real w,wk,wkm,ek
      real tnorm,ynorm,s,sm,sasum
      integer i1,j,j1,j2,k,kk,l
      logical lower
c***first executable statement  strco
      lower = job .eq. 0
c
c     compute 1-norm of t
c
      tnorm = 0.0e0
      do 10 j = 1, n
         l = j
         if (lower) l = n + 1 - j
         i1 = 1
         if (lower) i1 = j
         tnorm = amax1(tnorm,sasum(l,t(i1,j),1))
   10 continue

      ek = 1.0e0
      do 20 j = 1, n
         z(j) = 0.0e0
   20 continue
      do 100 kk = 1, n
         k = kk
         if (lower) k = n + 1 - kk
         if (z(k) .ne. 0.0e0) ek = sign(ek,-z(k))
         if (abs(ek-z(k)) .le. abs(t(k,k))) go to 30
            s = abs(t(k,k))/abs(ek-z(k))
            Call sscal(n,s,z,1)
            ek = s*ek
   30    continue
         wk = ek - z(k)
         wkm = -ek - z(k)
         s = abs(wk)
         sm = abs(wkm)
         if (t(k,k) .eq. 0.0e0) go to 40
            wk = wk/t(k,k)
            wkm = wkm/t(k,k)
         go to 50
   40    continue
            wk = 1.0e0
            wkm = 1.0e0
   50    continue
         if (kk .eq. n) go to 90
            j1 = k + 1
            if (lower) j1 = 1
            j2 = n
            if (lower) j2 = k - 1
            do 60 j = j1, j2
               sm = sm + abs(z(j)+wkm*t(k,j))
               z(j) = z(j) + wk*t(k,j)
               s = s + abs(z(j))
   60       continue
```

```
            if (s .ge. sm) go to 80
               w = wkm - wk
               wk = wkm
               do 70 j = j1, j2
                  z(j) = z(j) + w*t(k,j)
   70          continue
   80       continue
   90    continue
         z(k) = wk
  100 continue
      s = 1.0e0/sasum(n,z,1)
      Call sscal(n,s,z,1)
c
      ynorm = 1.0e0
c
c     solve t*z = y
c
      do 130 kk = 1, n
         k = n + 1 - kk
         if (lower) k = kk
         if (abs(z(k)) .le. abs(t(k,k))) go to 110
            s = abs(t(k,k))/abs(z(k))
            Call sscal(n,s,z,1)
            ynorm = s*ynorm
  110    continue
         if (t(k,k) .ne. 0.0e0) z(k) = z(k)/t(k,k)
         if (t(k,k) .eq. 0.0e0) z(k) = 1.0e0
         i1 = 1
         if (lower) i1 = k + 1
         if (kk .ge. n) go to 120
            w = -z(k)
            Call saxpy(n-kk,w,t(i1,k),1,z(i1),1)
  120    continue
  130 continue
c     make znorm = 1.0
      s = 1.0e0/sasum(n,z,1)
      Call sscal(n,s,z,1)
      ynorm = s*ynorm
c
      if (tnorm .ne. 0.0e0) rcond = ynorm/tnorm
      if (tnorm .eq. 0.0e0) rcond = 0.0e0
      return
      end

c*****************************************************************

      real function sasum(n,sx,incx)
c     ------------------------------
c=====================================================================
c        n  number of elements in input vector(s)
c       sx  single precision vector with n elements
c     incx  storage spacing between elements of sx
c
c***references  lawson c.l., hanson r.j., kincaid d.r., krogh f.t.,
c                 *basic linear algebra subprograms for fortran usage*,
c                 algorithm no. 539, transactions on mathematical
c                 software, volume 5, number 3, september 1979, 308-323
c=====================================================================
      real sx(1)
c***first executable statement  sasum
      sasum = 0.0e0
      if(n.le.0)return
      if(incx.eq.1)goto 20
c
c  code for increments not equal to 1.
c
      ns = n*incx
          do 10 i=1,ns,incx
          sasum = sasum + abs(sx(i))
   10     continue
      return
c
c  code for increments equal to 1.
c  clean-up loop so remaining vector length is a multiple of 6.
```

```
c
   20 m = mod(n,6)
      if( m .eq. 0 ) go to 40
      do 30 i = 1,m
        sasum = sasum + abs(sx(i))
   30 continue
      if( n .lt. 6 ) return
   40 mp1 = m + 1
      do 50 i = mp1,n,6
        sasum = sasum + abs(sx(i)) + abs(sx(i + 1)) + abs(sx(i + 2))
     1  + abs(sx(i + 3)) + abs(sx(i + 4)) + abs(sx(i + 5))
   50 continue
      return
      end

c*********************************************************************

      subroutine saxpy(n,sa,sx,incx,sy,incy)
c     ----------------------------------
c=====================================================================
c        n  number of elements in input vector(s)
c       sa  single precision scalar multiplier
c       sx  single precision vector with n elements
c     incx  storage spacing between elements of sx
c       sy  single precision vector with n elements
c     incy  storage spacing between elements of sy
c
c references  lawson c.l., hanson r.j., kincaid d.r., krogh f.t.,
c        *basic linear algebra subprograms for fortran usage*,
c         algorithm no. 539, transactions on mathematical
c       software, volume 5, number 3, september 1979, 308-323
c
c=====================================================================
      real sx(1),sy(1),sa
c***first executable statement  saxpy
      if(n.le.0.or.sa.eq.0.e0) return
      if(incx.eq.incy) if(incx-1) 5,20,60
    5 continue
c
c  code for nonequal or nonpositive increments.
c
      ix = 1
      iy = 1
      if(incx.lt.0)ix = (-n+1)*incx + 1
      if(incy.lt.0)iy = (-n+1)*incy + 1
      do 10 i = 1,n
        sy(iy) = sy(iy) + sa*sx(ix)
        ix = ix + incx
        iy = iy + incy
   10 continue
      return
c
c  code for both increments equal to 1
c  clean-up loop so remaining vector length is a multiple of 4.
c
   20 m = mod(n,4)
      if( m .eq. 0 ) go to 40
      do 30 i = 1,m
        sy(i) = sy(i) + sa*sx(i)
   30 continue
      if( n .lt. 4 ) return
   40 mp1 = m + 1
      do 50 i = mp1,n,4
        sy(i) = sy(i) + sa*sx(i)
        sy(i + 1) = sy(i + 1) + sa*sx(i + 1)
        sy(i + 2) = sy(i + 2) + sa*sx(i + 2)
        sy(i + 3) = sy(i + 3) + sa*sx(i + 3)
   50 continue
      return
c
c  code for equal, positive, nonunit increments.
c
   60 continue
      ns = n*incx
```

```
          do 70 i=1,ns,incx
          sy(i) = sa*sx(i) + sy(i)
   70     continue
      return
      end

c*************************************************************

      subroutine sscal(n,sa,sx,incx)
c     ----------------------------
c=====================================================================
c        n  number of elements in input vector(s)
c       sa  single precision scale factor
c       sx  single precision vector with n elements
c     incx  storage spacing between elements of sx

c       sx  single precision result (unchanged if n .le. 0)
c
c     replace single precision sx by single precision sa*sx.
c     for i = 0 to n-1, replace sx(1+i*incx) with  sa * sx(1+i*incx)
c  references  lawson c.l., hanson r.j., kincaid d.r., krogh f.t.,
c         *basic linear algebra subprograms for fortran usage*,
c          algorithm no. 539, transactions on mathematical
c       software, volume 5, number 3, september 1979, 308-323
c=====================================================================
      real sa,sx(1)
c***first executable statement  sscal
      if(n.le.0)return
      if(incx.eq.1)goto 20
c
c        code for increments not equal to 1.
c
      ns = n*incx
          do 10 i = 1,ns,incx
          sx(i) = sa*sx(i)
   10     continue
      return
c
c        code for increments equal to 1.
c        clean-up loop so remaining vector length is a multiple of 5.
c
   20 m = mod(n,5)
      if( m .eq. 0 ) go to 40
      do 30 i = 1,m
        sx(i) = sa*sx(i)
   30 continue
      if( n .lt. 5 ) return
   40 mp1 = m + 1
      do 50 i = mp1,n,5
        sx(i) = sa*sx(i)
        sx(i + 1) = sa*sx(i + 1)
        sx(i + 2) = sa*sx(i + 2)
        sx(i + 3) = sa*sx(i + 3)
        sx(i + 4) = sa*sx(i + 4)
   50 continue
      return
      end

c**********************************************************************

      subroutine svdcmp(a,m,n,mp,np,w,v)
c=====================================================================
c   Performs Singular Value Decomposition (see Press et al. 1992)
c=====================================================================
      integer m,mp,n,np,nmax
      real a(mp,np),v(np,np),w(np)
      parameter (nmax=500)
cu    uses pythag
      integer i,its,j,jj,k,l,nm
      real anorm,c,f,g,h,s,scale,x,y,z,rv1(nmax),pythag
      g=0.0
      scale=0.0
      anorm=0.0
      do 25 i=1,n
```

```
        l=i+1
        rv1(i)=scale*g
        g=0.0
        s=0.0
        scale=0.0
        if(i.le.m)then
          do 11 k=i,m
            scale=scale+abs(a(k,i))
11        continue
          if(scale.ne.0.0)then
            do 12 k=i,m
              a(k,i)=a(k,i)/scale
              s=s+a(k,i)*a(k,i)
12          continue
            f=a(i,i)
            g=-sign(sqrt(s),f)
            h=f*g-s
            a(i,i)=f-g
            do 15 j=l,n
              s=0.0
              do 13 k=i,m
                s=s+a(k,i)*a(k,j)
13            continue
              f=s/h
              do 14 k=i,m
                a(k,j)=a(k,j)+f*a(k,i)
14            continue
15          continue
            do 16 k=i,m
              a(k,i)=scale*a(k,i)
16          continue
          endif
        endif
        w(i)=scale *g
        g=0.0
        s=0.0
        scale=0.0
        if((i.le.m).and.(i.ne.n))then
          do 17 k=l,n
            scale=scale+abs(a(i,k))
17        continue
          if(scale.ne.0.0)then
            do 18 k=l,n
              a(i,k)=a(i,k)/scale
              s=s+a(i,k)*a(i,k)
18          continue
            f=a(i,l)
            g=-sign(sqrt(s),f)
            h=f*g-s
            a(i,l)=f-g
            do 19 k=l,n
              rv1(k)=a(i,k)/h
19          continue
            do 23 j=l,m
              s=0.0
              do 21 k=l,n
                s=s+a(j,k)*a(i,k)
21            continue
              do 22 k=l,n
                a(j,k)=a(j,k)+s*rv1(k)
22            continue
23          continue
            do 24 k=l,n
              a(i,k)=scale*a(i,k)
24          continue
          endif
        endif
        anorm=max(anorm,(abs(w(i))+abs(rv1(i))))
25    continue
      do 32 i=n,1,-1
        if(i.lt.n)then
          if(g.ne.0.0)then
            do 26 j=l,n
              v(j,i)=(a(i,j)/a(i,l))/g
```

```
26          continue
            do 29 j=l,n
              s=0.0
              do 27 k=l,n
                s=s+a(i,k)*v(k,j)
27            continue
              do 28 k=l,n
                v(k,j)=v(k,j)+s*v(k,i)
28            continue
29          continue
          endif
          do 31 j=l,n
            v(i,j)=0.0
            v(j,i)=0.0
31        continue
        endif
        v(i,i)=1.0
        g=rv1(i)
        l=i
32    continue
      do 39 i=min(m,n),1,-1
        l=i+1
        g=w(i)
        do 33 j=l,n
          a(i,j)=0.0
33      continue
        if(g.ne.0.0)then
          g=1.0/g
          do 36 j=l,n
            s=0.0
            do 34 k=l,m
              s=s+a(k,i)*a(k,j)
34          continue
            f=(s/a(i,i))*g
            do 35 k=i,m
              a(k,j)=a(k,j)+f*a(k,i)
35          continue
36        continue
          do 37 j=i,m
            a(j,i)=a(j,i)*g
37        continue
        else
          do 38 j= i,m
            a(j,i)=0.0
38        continue
        endif
        a(i,i)=a(i,i)+1.0
39    continue
      do 49 k=n,1,-1
        do 48 its=1,30
          do 41 l=k,1,-1
            nm=l-1
            if((abs(rv1(l))+anorm).eq.anorm)  goto 2
            if((abs(w(nm))+anorm).eq.anorm)  goto 1
41        continue
1         c=0.0
          s=1.0
          do 43 i=l,k
            f=s*rv1(i)
            rv1(i)=c*rv1(i)
            if((abs(f)+anorm).eq.anorm) goto 2
            g=w(i)
            h=pythag(f,g)
            w(i)=h
            h=1.0/h
            c= (g*h)
            s=-(f*h)
            do 42 j=1,m
              y=a(j,nm)
              z=a(j,i)
              a(j,nm)=(y*c)+(z*s)
              a(j,i)=-(y*s)+(z*c)
42          continue
43        continue
```

```
2         z=w(k)
          if(l.eq.k)then
            if(z.lt.0.0)then
              w(k)=-z
              do 44 j=1,n
                v(j,k)=-v(j,k)
44            continue
            endif
            goto 3
          endif
          if(its.eq.30) pause 'no convergence in svdcmp'
          x=w(l)
          nm=k-1
          y=w(nm)
          g=rv1(nm)
          h=rv1(k)
          f=((y-z)*(y+z)+(g-h)*(g+h))/(2.0*h*y)
          g=pythag(f,1.0)
          f=((x-z)*(x+z)+h*((y/(f+sign(g,f)))-h))/x
          c=1.0
          s=1.0
          do 47 j=l,nm
            i=j+1
            g=rv1(i)
            y=w(i)
            h=s*g
            g=c*g
            z=pythag(f,h)
            rv1(j)=z
            c=f/z
            s=h/z
            f= (x*c)+(g*s)
            g=-(x*s)+(g*c)
            h=y*s
            y=y*c
            do 45 jj=1,n
              x=v(jj,j)
              z=v(jj,i)
              v(jj,j)= (x*c)+(z*s)
              v(jj,i)=-(x*s)+(z*c)
45          continue
            z=pythag(f,h)
            w(j)=z
            if(z.ne.0.0)then
              z=1.0/z
              c=f*z
              s=h*z
            endif
            f= (c*g)+(s*y)
            x=-(s*g)+(c*y)
            do 46 jj=1,m
              y=a(jj,j)
              z=a(jj,i)
              a(jj,j)= (y*c)+(z*s)
              a(jj,i)=-(y*s)+(z*c)
46          continue
47        continue
          rv1(l)=0.0
          rv1(k)=f
          w(k)=x
48      continue
3       continue
49    continue
      return
      end
c  (c) copr. 1986-92 numerical recipes software

c**********************************************************************

      function pythag(a,b)
c     --------------------
c=======================================================================
c     For use with SVDCMP (See Press et al. 1992)
c=======================================================================
```

```
      real a,b,pythag
      real absa,absb
      absa=abs(a)
      absb=abs(b)
      if(absa.gt.absb)then
         pythag=absa*sqrt(1.+(absb/absa)**2)
      else
         if(absa.eq.0)then
            pythag=0.
         else
            pythag=absb*sqrt(1.+(absa/absb)**2)
         endif
      endif
      return
      end
c  (c) copr. 1986-92 numerical recipes software

c**********************************************************************

      function urand()
c=======================================================================
c     return the next pseudo-random deviate from a sequence which is
c     uniformly distributed in the interval [0,1]
c
c     uses the function ran0, the "minimal standard" random number
c     generator of park and miller (comm. acm 31, 1192-1201, oct 1988;
c     comm. acm 36 no. 7, 105-110, july 1993).
c=======================================================================
      implicit none
      real     urand, ran0
      integer  iseed
      external ran0
      common /rnseed/ iseed
c
      urand = ran0( iseed )
      return
      end
c**********************************************************************

      subroutine rninit( seed )
c=======================================================================
c     initialize random number generator urand with given seed
c=======================================================================
      implicit none
      integer seed, iseed
c     common block to communicate with urand
      common /rnseed/ iseed
c
c     set the seed value
      iseed = seed
      if(iseed.le.0) iseed=123456
      return
      end

c**********************************************************************

      function ran0( seed )
c=======================================================================
c     "minimal standard" pseudo-random number generator of park and
c     miller.  returns a uniform random deviate r s.t. 0 < r < 1.0.
c     set seed to any non-zero integer value to initialize a sequence,
c     then do not change seed between calls for successive deviates
c     in the sequence.
c
c     references:
c        park, s. and miller, k., "random number generators: good ones
c           are hard to find", comm. acm 31, 1192-1201 (oct. 1988)
c        park, s. and miller, k., in "remarks on choosing and imple-
c           menting random number generators", comm. acm 36 no. 7,
c           105-110 (july 1993)
c=======================================================================
      implicit  none
      integer   seed, a, m, q, r, j
      real      ran0, scale, eps, rnm
```

```
      parameter (a=48271,m=2147483647,q=44488,r=3399)
      parameter (scale=1./m,eps=1.2e-7,rnmx=1.-eps)

      j = seed/q
      seed = a*(seed-j*q)-r*j
      if (seed .lt. 0) seed = seed+m
      ran0 = min(seed*scale,rnmx)
      return
      end

c**********************************************************************

      subroutine rqsort(n,a,p)
c======================================================================
c     return integer array p which indexes array a in increasing order.
c     array a is not disturbed.  the quicksort algorithm is used.
c
c     b. g. knapp, 86/12/23
c
c     reference: n. wirth, algorithms and data structures,
c     prentice-hall, 1986
c======================================================================
      implicit none
      integer   n, p(n), lgn, q
      integer   stackl(lgn),stackr(lgn),s,t,l,m,r,i,j
      real      a(n), x
      parameter (lgn=32, q=11)

c        (lgn = log base 2 of maximum n;
c         q = smallest subfile to use quicksort on)

c  initialize the stack
      stackl(1)=1
      stackr(1)=n
      s=1

c  initialize the pointer array
      do 1 i=1,n
         p(i)=i
    1 continue
    2 if (s.gt.0) then
         l=stackl(s)
         r=stackr(s)
         s=s-1
    3    if ((r-l).lt.q) then
c  use straight insertion
            do 6 i=l+1,r
               t = p(i)
               x = a(t)
               do 4 j=i-1,l,-1
                  if (a(p(j)).le.x) goto 5
                  p(j+1) = p(j)
    4          continue
               j=l-1
    5          p(j+1) = t
    6       continue
         else
c  use quicksort, with pivot as median of a(l), a(m), a(r)
            m=(l+r)/2
            t=p(m)
            if (a(t).lt.a(p(l))) then
               p(m)=p(l)
               p(l)=t
               t=p(m)
            endif
            if (a(t).gt.a(p(r))) then
               p(m)=p(r)
               p(r)=t
               t=p(m)
               if (a(t).lt.a(p(l))) then
                  p(m)=p(l)
                  p(l)=t
                  t=p(m)
               endif
```

```
            endif
c  partition
            x=a(t)
            i=l+1
            j=r-1
    7       if (i.le.j) then
    8          if (a(p(i)).lt.x) then
                  i=i+1
                  goto 8
               endif
    9          if (x.lt.a(p(j))) then
                  j=j-1
                  goto 9
               endif
               if (i.le.j) then
                  t=p(i)
                  p(i)=p(j)
                  p(j)=t
                  i=i+1
                  j=j-1
               endif
               goto 7
            endif
c  stack the larger subfile
            s=s+1
            if ((j-l).gt.(r-i)) then
               stackl(s)=l
               stackr(s)=j
               l=i
            else
               stackl(s)=i
               stackr(s)=r
               r=j
            endif
            goto 3
         endif
         goto 2
      endif
      return
      end

c**********************************************************************
c**********************************************************************
```

References

Acton, L. W., Finch, M. L., Gilbreth, C. W., Culhane, J. L., Bentley, R. D., Bowles, J. A., Guttridge, P., Gabriel, A. H., Firth, J. G., Hayes, R. W. 1980, Solar Phys., 65, 53

Almleaky, Y. M., Brown, J. C., Sweet, P. A. 1989, A&A, 224, 328

Arnaud, M., Rothenflug, R. 1985, A&AS, 60, 425

Basu, S., Christensen-Dalsgaard, J., Thompson, M. J. 1997, A&A, 321, 634

Bertero, M. 1997, Private Communication

Beyer, W. H. 1991, CRC Standard Mathematical Tables and Formulae, CRC Press, Boca Raton, Florida, 29th edition

Brage, T., Judge, P. G., Brekke, P. 1996, ApJ, 464, 1030

Bray, R. J., Loughhead, R. E., Durrant, C. J. 1984, The solar granulation, Cambridge University Press, Cambridge, UK

Brekke, P., Kjeldseth-Moe, O., Brynildsen, N., Maltby, P., Haugan, S. V. H., Harrison, R. A., Thompson, W. T., Pike, C. D. 1997, Solar Phys., 170, 163

Brickhouse, N. S., Raymond, J. C., Smith, B. W. 1995, ApJS, 97, 551

Brosius, J. W., Davila, J. M., Thomas, R. J., Monsignori-Fossi, B. C. 1996, ApJS, 106, 143

Brown, J. C. 1971, Solar Phys., 18, 489

Brown, J. C. 1978, ApJ, 225, 1076

Brown, J. C., Dwivedi, B. N., Sweet, P. A., Almleaky, Y. M. 1991, A&A, 249, 277

Brown, J. C., McArthur, G. A., Barrett, R. K., McIntosh, S. W., Emslie, A. G. 1998, Solar Phys., 179(2), 379

Brown, R., Lang, J. (eds.) 1988, Astrophysical and Laboratory Spectroscopy: Proceedings of the Thirty-Third Scottish Universities Summer School in Physics., European Solar Observatory

Brynildsen, N. 1994, Profile fitting to CDS/SUMER data : CDS software note No. 21, Technical report, University of Oslo

Chae, J., Yun, H. S., Poland, A. I. 1997, ApJ, 480, 817

Charbonneau, P. 1995, ApJS, 101, 309

Charbonneau, P. 1998, Private Communication

Charbonneau, P., Knapp, B. 1996, A User's Guide to PIKAIA 1.0 : NCAR Technical Note 418+IA

Christensen-Dalsgaard, J., Hansen, P. C., Thompson, M. J. 1993, MNRAS, 264, 541

Cline, A. K., Moler, C. B., Stewart, G. W., Wilkinson, J. H. 1979, SIAM J. Num. Anal., 16, 368

Craig, I. J. D., Brown, J. C. 1976, A&A, 49, 239

Craig, I. J. D., Brown, J. C. 1986, Inverse Problems in Astronomy : A Guide to Inversion Strategies of Remotely Sensed Data, Adam Hilger, Bristol, UK.

Darwin, C. 1859, On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life, J. Murray: London

Davis, L. (ed.) 1991, Handbook of Genetic Algorithms, Van Nostrand Reinhold

Dere, K., Mason, H. E. 1981, in F. Q. Orall (ed.), Solar Active Regions, Boulder: Colorado Associated University, 129

Dere, K. P., Landi, E., Mason, H. E., Monsignori-Fossi, B. C., Young, P. R. 1997, A&AS, 125, 149

Diver, D. A. 1995, Eur. J. Phys., 16, 211

Diver, D. A., Ireland, D. G. 1997, Nucl. Instr. and Meth. A, 399, 414

Doschek, G. A. 1984, ApJ, 279, 446

Doschek, G. A. 1987, in R. G. Athay, D. S. Spicer (eds.), Theoretical Problems in High Resolution Solar Physics II, NASA CP 2483, 37

Edlén, B. 1941, Arkiv för Matematik, Astronomi, o. Fysik, 28B(1), 1

Edlén, B. 1943, Zeitschrift für Astrofysik, 22, 30

Fleck, B., Domingo, V., Poland, A. I. 1995, The SOHO mission, Dordrecht: Kluwer

Fludra, A., Schmelz, J. T. 1995, ApJ, 447, 936

Foukal, P. 1990, Solar Astrophysics, Wiley and Sons, New York

Gabriel, A. H., Jordan, C. 1969, MNRAS, 145, 241

Gabriel, A. H., Jordan, C. 1971, Case Studies in Atomic Collision Physics, Chapt. 4, 210–291, Elsevier:North-Holland

Goldberg, D. A. 1989, Genetic Algorithms in Search of Optimization and Machine Learning, Addison Wesley

Golub, G. H., Van Loan, C. F. 1989, Matrix Computations, Johns Hopkins University Press, Baltimore, MD., 2nd edition

Golub, L., Pasachoff, J. M. 1997, The Solar Corona, Cambridge University Press

Goncharskii, A. V., Leonov, A. S., Yagoda, A. G. 1972, Zh. Vychisl. Mat. Fiz., 12(6), 1592

Griffiths, N. W., Jordan, C. 1998, ApJ, 497, 883

Grotrian, W. 1928, Graphische Darstellung der Spektren von Atomen un Ionen mit Ein, Zwei, und Drei Valenzelektronen, Springer, Berlin

Hansen, P. C. 1994, Inverse Problems, 10, 895

Harrison, R. A., Sawyer, E. C., Carter, M. K., Cruise, A. M., Cutler, R. M., Fludra, A., Hayes, R. W., Kent, B. J., Lang, J., Parker, D. J., Payne, J., Pike, C. D., Peskett, S. C., Richards, A. G., Culhane, J. L., Norman, K., Breeveld, A. A., Breeveld, E. R., Janabi, K. F. A., McCalden, A. J., Parkinson, J. H., Self, D. G., Thomas, P. D., Poland, A. I., Thomas, R. J., Thompson, W. T., Kjeldseth-Moe, O., Brekke, P., Karud, J., Maltby, P., Aschenbach, B., Bräuninger, H., Kühne, M., Hollandt, J., Sigmund, O. H. W., Huber, M. C. E., Gabriel, A. H., Mason, H. E., Bromage, B. J. I. 1995, Solar Phys., 162, 233

Harrison, R. A., Thompson, A. M. 1991, Intensity Integral Inversion Techniques: a Study in Preparation for the SOHO Mission, Technical Report RAL-91-092, Rutherford Appleton Laboratory

Holland, J. H. 1962, J. Assoc. Comp. Mach., 3, 297

Hubeny, V., Judge, P. G. 1995, ApJ, 448, L61

Jacobs, V. L., Davis, J., Kepple, P. C., Blaha, M. 1977, ApJ, 211, 605

Jacobs, V. L., Davis, J., Robertson, J. E., Blaha, M., Cain, J., Davis, M. 1980, ApJ, 239, 1119

Jefferies, J. T., Orrall, F. Q., Zirker, J. B. 1972a, Solar Phys., 22, 307

Jefferies, J. T., Orrall, F. Q., Zirker, J. B. 1972b, Solar Phys., 22, 317

Jin, Q., Hou, Z. 1997, Inverse Problems, 13, 815

Jordan, C. 1969, MNRAS, 142, 501

Jordan, C. 1970, MNRAS, 148, 17

Jordan, C., Ayres, T. R., Brown, A., Linsky, J. L., Simon, T. 1987, MNRAS, 225, 903

Judge, P. G., Hansteen, V., Wikstol, O., Wilhelm, K., Schühle, U., Moran, T. 1998, ApJ, 502, 981

Judge, P. G., Hubeny, V., Brown, J. C. 1997, ApJ, 475, 275

Judge, P. G., Woods, T. N., Brekke, P., Rottman, G. J. 1995, ApJ, 455, L85

Kashyap, V., Drake, J. J. 1998, ApJ, 503, 450

Laming, J. M., Feldman, U., Schuehle, U., Lemaire, P., Curdt, W., Wilhelm, K. 1997, ApJ, 485, 911

Landi, E., Landini, M. 1997, A&A, 327, 1230

Lanzafame, A. C., Brooks, D. H., Summers, H. P., Thomas, R. J., Thompson, A. M. 1998, A&A, In preparation

Litwin, C., Rosner, R. 1993, ApJ, 412, 375

Lloyd-Hart, M., Angel, J. R. P., Groesbeck, T. D., Martinez, T., Jacobsen, B. P., McLeod, B. A., McCarthy, D. W., Hooper, E. J., Hege, E. K., Sandler, D. G. 1998, ApJ, 493, 950

Louis, A. K. 1996, Inverse Problems, 12, 175

Mariska, J. T. 1992, The Solar Transition Region, Cambridge University Press, Cambridge UK

Mason, H. E. 1991, Adv. Space Res., 11, 293

Mason, H. E., Monsignori-Fossi, B. C. 1994, A&ARv, 6, 123

McIntosh, S. W. 1998, Are we getting the most out of DEM ?, Oral presentation

McIntosh, S. W., Brown, J. C., Judge, P. G. 1998a, A&A, 333, 333

McIntosh, S. W., Diver, D. A., Judge, P. G., Charbonneau, P., Ireland, J., Brown, J. C. 1998b, A&AS, 132, 145

Menzel, D. H., Aller, L. H., Hebb, M. H. 1941, ApJ, 93, 230

Michalewicz, Z. 1994, Genetic Algorithms + Data Structures = Evolution Programs, Springer, New York

Mitchell, M. 1996, An Introduction to Gentic Algorithms, MIT Press, Cambridge, Mass.

Moseley, H. G. J. 1913, Phil. Mag., 1024

Munro, R. H., Dupree, A. K., Withbroe, G. L. 1971, Solar Phys., 19, 347

Neupert, W. M., Gates, W., M., S., Young, R. 1962, ApJ, 149, L79

Orrall, F. Q. 1981, Solar active regions: A monograph from Skylab Solar Workshop III, Skylab Solar Workshop III, NASA

Parker, E. N. 1988, ApJ, 330, 474

Petrov, A. P., Khovanskii, A. V. 1973, USSR Comp. Math. Math. Phys., 13, 17

Phillips, D. L. 1962, J. Ass. Comp. Mach., 9, 84

Pottasch, S. R. 1964, Space Sci. Rev., 3, 816

Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. 1992, Numerical Recipes in Fortran: Second Edition, Cambridge University Press

Rust, B. W., Burrus, W. R. 1972, Mathematical Programming and the Numerical Solution of Linear Equations, Elsevier, Amsterdam

Savin, D. W., Gardner, L. D., Reisenfeld, D. B., Young, A. R., Kohl, J. L. 1995, Phys. Rev. A, 51(3), 2161

Schou, J., Christensen-Dalsgaard, J., Thompson, M. J. 1994, ApJ, 433, 389

Seely, J. F., Feldman, U., Schühle, U., Wilhelm, K., Curdt, W., Lemaire, P. 1997, ApJ, 484, 87

Sneddon, I. N. 1972, The Use of Integral Transforms, McGraw-Hill, New York

Stenflo, J. O. 1994, Solar magnetic fields: polarized radiation diagnostics, Dordrecht: Kluwer, Boston

Stix, M. 1989, The Sun: An Introduction, Springer, Berlin

Sturrock, P. A. 1980, Solar flares: A monograph from Skylab Solar Workshop II, Skylab Solar Workshop II, NASA

Sturrock, P. A. 1985, Physics of the Sun, University of Chicago Press

Thomas, R. J., Neupert, W. M. 1994, ApJS, 91, 461

Thomas, R. N., Athay, R. G. 1961, Physics of the Solar Chromosphere, Interscience, New York

Thompson, A. M. 1990, A&A, 240, 209

Thompson, A. M. 1991, in A. M. Thompson, R. A. Harrison (eds.), Integral Intensity Inversion Techniques: a study in preparation for the SOHO mission, No. RAL91-092, Rutherford Appleton Laboratory

Tichonov, A. N. 1963, Soviet Maths - *Dokl.*, 4, 1035

Twomey, S. 1963, J. Ass. Comp. Mach., 10, 97

Vernazza, J. E., Avrett, E. H., Loeser, R. 1981, ApJS, 45, 635

Whitelaw, T. A. 1983, An Introduction to Linear Algebra, Blackie, London

Wikstol, O., Judge, P. G., Hansteen, V. 1997, ApJ, 483, 972

Wikstol, O., Judge, P. G., Hansteen, V. 1998, ApJ, 501, 895

Wilhelm, K., Curdt, W., Marsch, E., Schühle, U., Lemaire, P., Gabriel, A. H., Vial, J. C., Grewing, M., Huber, M. C., Jordan, S. D., Poland, A. I., Thomas, R. J., Kühne, M., Timothy, J. G., Hassler, D. M., Siegmund, O. H. 1995, Solar Phys., 162, 189

Zirin, H. 1988, Astrophysics of the Sun, Cambridge University Press, Cambridge, Great Britain

Zirker, J. B. (ed.) 1977, Coronal Holes and High Speed Wind Streams, Colorado Associated University Press, Boulder, CO

Zirker, J. B. 1993, Solar Phys., 148, 43